

ultimo theme

Extremely customizable Magento theme by Infortis

User Guide

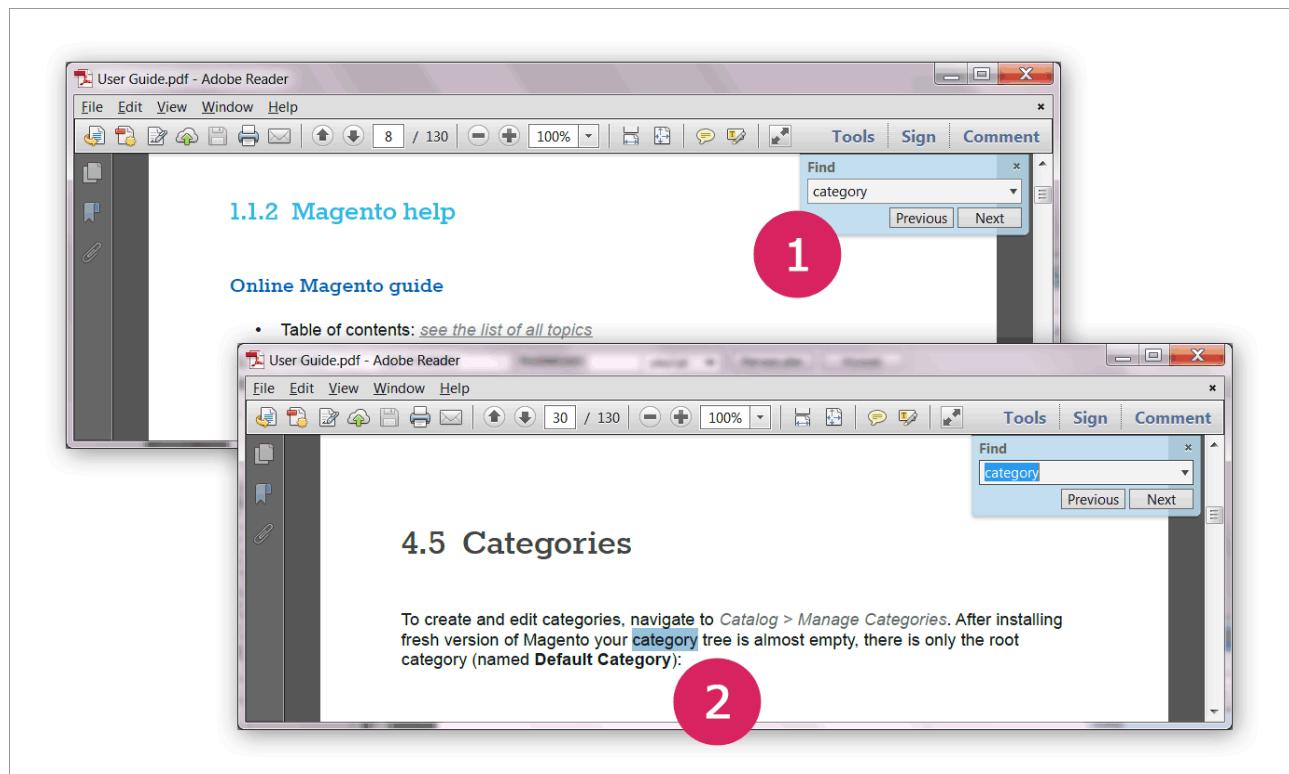
Copyright © 2012-2017 Infortis. All rights reserved.

Document updated: September 03, 2017

How to use this document

Please read this user guide carefully, it will help you eliminate most of potential problems with incorrect configuration of the theme and Magento.

Use the search tool in your PDF reader (click **CTRL+F** in most readers) to quickly find all the keywords which you are looking for:



This document is organized as follows:

- Chapters **1. FAQ + Quick Start** and **2. Help & Support** provide an overview of this theme, some basic information about support and links to many useful websites about Magento.
- Chapter **3. Installation** describes theme installation process.

- Chapter **4. Magento configuration** describe selected Magento features.
- Chapter **5. Customization** shows how to customize the theme. Subchapter **5.2** describes Magento themes structure and some basic concepts which you need to know when starting to work with Magento.
- Chapter **6. Theme features and elements** provides a description of selected elements of this theme, such as product sliders, home page, built-in static blocks, useful CSS classes etc.
- Chapters from **7.** to **13.** – specify a list of all theme settings available in the theme admin panel. **Each chapter refers to specific section of theme admin panel.**
- Chapter **14. Troubleshooting** describes the most common problems which you may encounter after installation or during theme and Magento configuration.

All chapters are listed in the **Table of Contents** below.

Table of Contents

| | |
|--|-----------|
| 1. FAQ + Quick Start..... | 11 |
| 1.1 Help & support..... | 11 |
| 1.2 License..... | 12 |
| 1.3 Important things you need to know before you start using Magento..... | 13 |
| 1.4 Installation..... | 13 |
| 1.5 After installation..... | 13 |
| 1.6 Fresh Magento installation..... | 15 |
| 1.7 Theme customization..... | 15 |
| 2. Help & Support..... | 16 |
| 2.1 References..... | 16 |
| 2.1.1 Magento installation..... | 16 |
| 2.1.2 Magento help..... | 17 |
| 2.2 Support policy..... | 19 |
| 2.3 Third-party extensions..... | 21 |
| 3. Installation..... | 23 |
| 3.1 Installation of demo store..... | 23 |
| 3.2 Installation for live Magento store..... | 24 |
| 3.2.1 Prepare your Magento..... | 24 |
| 3.2.2 Install..... | 26 |
| 3.2.3 Import sample blocks and pages..... | 29 |
| 3.3 Problems after installation..... | 30 |
| 4. Magento configuration..... | 32 |
| 4.1 Caching system..... | 33 |
| 4.1.1 Flush cache..... | 33 |
| 4.1.2 Disable cache..... | 34 |
| 4.1.3 Enable cache..... | 34 |
| 4.1.4 Refresh cache..... | 35 |
| 4.2 Index management..... | 36 |
| 4.3 How to enable the theme..... | 37 |
| 4.4 Logo..... | 39 |
| 4.4.1 Sticky logo..... | 39 |
| 4.5 Categories..... | 41 |
| 4.5.1 Category info..... | 44 |
| 4.5.2 Find category ID..... | 44 |
| 4.5.3 Landing pages..... | 44 |
| 4.5.4 Depth of the menu..... | 45 |
| 4.5.5 Layered Navigation ("Shop By" filters)..... | 45 |
| 4.5.6 Categories inside Layered Navigation..... | 46 |
| 4.6 Products..... | 47 |

| | |
|--|-----------|
| 4.6.1 Position of: Add to Cart button and product options..... | 47 |
| 4.6.2 "Only X left" stock indicator..... | 48 |
| 4.7 Attributes..... | 49 |
| 4.7.1 Attribute Management | 49 |
| 4.7.2 Attribute Sets..... | 52 |
| 4.8 Content Management System (CMS)..... | 54 |
| 4.8.1 Add blocks to selected pages..... | 54 |
| 4.8.2 Static blocks..... | 57 |
| 4.8.3 CMS markup tags..... | 59 |
| 4.8.3.A Store URL..... | 59 |
| 4.8.3.B Media URL..... | 60 |
| 4.8.3.C Block ID..... | 61 |
| 4.8.4 WYSIWYG editor..... | 61 |
| 4.9 Translation / Localization..... | 63 |
| 4.9.1 Interface translation..... | 63 |
| 4.9.2 Language flag..... | 64 |
| 5. Customization..... | 65 |
| 5.1 How to modify the theme..... | 66 |
| 5.1.1 Customize images..... | 66 |
| 5.1.2 Override default CSS styles..... | 67 |
| 5.1.3 Modify template files..... | 69 |
| 5.1.4 Create custom sub-theme..... | 70 |
| 5.2 Magento theme customization..... | 71 |
| 5.2.1 Design packages..... | 71 |
| 5.2.2 Theme structure..... | 72 |
| 5.2.3 Fallback mechanism..... | 73 |
| 5.2.4 Examples of custom sub-theme..... | 75 |
| 5.2.5 Summary..... | 78 |
| 6. Theme features and elements..... | 79 |
| 6.1 Home page and CMS pages..... | 80 |
| 6.1.1 Default pages..... | 80 |
| 6.1.1.A Selecting a home page..... | 80 |
| 6.1.2 Creating a page..... | 80 |
| 6.1.2.A Page layout..... | 81 |
| 6.1.2.B Page content..... | 81 |
| 6.1.2.C Full width page..... | 82 |
| 6.1.2.C.1 Maximum width of the content..... | 82 |
| 6.1.2.C.2 Content with full width background..... | 83 |
| 6.1.3 Home page blocks..... | 84 |
| 6.1.3.A Image slideshow..... | 84 |
| 6.1.3.B Banners..... | 84 |
| 6.1.3.C Grid/List of Products..... | 84 |
| 6.1.4 Featured Products Slider..... | 86 |
| 6.1.4.A List of parameters..... | 86 |
| 6.1.4.B Examples of shortcodes..... | 90 |
| 6.1.4.C Display any set of products using the slider..... | 92 |
| 6.1.5 New Products Slider..... | 94 |
| 6.2 Built-in CMS blocks..... | 98 |
| 6.2.1 Page..... | 100 |
| 6.2.2 Header..... | 102 |
| 6.2.3 Footer..... | 107 |
| 6.2.4 Category view..... | 110 |
| 6.2.5 Product page..... | 111 |
| 6.2.6 Shopping cart..... | 113 |
| 6.2.7 Cart drop-down..... | 113 |

| | |
|---|------------|
| 6.3 Pages and elements..... | 114 |
| 6.3.1 Product page..... | 114 |
| 6.3.1.A Social bookmarks (AddThis)..... | 114 |
| 6.3.2 Drop-down cart..... | 114 |
| 6.4 Grid system..... | 115 |
| 6.4.1 Bootstrap grid overview..... | 115 |
| 6.4.2 Grid examples..... | 117 |
| 6.4.3 Additional utility classes for grid system..... | 118 |
| 6.4.4 Old 12-column grid system (deprecated)..... | 119 |
| 6.5 CMS components..... | 124 |
| 6.5.1 CSS utility classes..... | 124 |
| 6.5.1.A General utility classes..... | 124 |
| 6.5.1.B Hide/show content depending on screen size..... | 126 |
| 6.5.1.C Old classes to hide elements (deprecated)..... | 128 |
| 6.5.2 Collapsible blocks..... | 129 |
| 6.5.3 Icons..... | 130 |
| 6.5.3.A Basic icons..... | 130 |
| 6.5.3.A.1 Add CSS styles to icons..... | 131 |
| 6.5.3.A.2 Size..... | 131 |
| 6.5.3.A.3 List icons..... | 132 |
| 6.5.3.A.4 Pulled icons and borders..... | 132 |
| 6.5.3.A.5 Spinning icons..... | 132 |
| 6.5.3.A.6 Rotated and flipped..... | 132 |
| 6.5.3.A.7 Icons inside buttons..... | 133 |
| 6.5.3.B Iconbox (icon with background color)..... | 133 |
| 6.5.3.B.1 Change color on mouse hover..... | 134 |
| 6.5.3.B.2 Size..... | 134 |
| 6.5.3.B.3 Shape..... | 134 |
| 6.5.3.B.4 Iconbox effects..... | 135 |
| 6.5.3.B.5 Image file iconbox..... | 135 |
| 6.5.3.C Social icons..... | 136 |
| 6.5.3.D Letters, numbers and other characters as icons..... | 138 |
| 6.5.3.E Font Awesome icons..... | 138 |
| 6.5.4 Icon + text..... | 140 |
| 6.5.4.A Simple block with icon..... | 140 |
| 6.5.4.B Change iconbox color on mouse hover..... | 141 |
| 6.5.4.C Indented block..... | 141 |
| 6.5.4.D Centered block..... | 142 |
| 6.5.5 Banners..... | 143 |
| 6.5.5.A Image banner..... | 143 |
| 6.5.5.A.1 How to get the image..... | 143 |
| 6.5.5.B Banner with hyperlink..... | 144 |
| 6.5.5.C Banner with image hover effect..... | 145 |
| 6.5.5.D Banner with color overlay..... | 145 |
| 6.5.5.E Banner with caption..... | 147 |
| 6.5.5.E.1 Inline styles..... | 147 |
| 6.5.5.E.2 Remove caption background color..... | 148 |
| 6.5.5.E.3 Caption with fade in/out effect..... | 148 |
| 6.5.5.E.4 Center caption..... | 149 |
| 6.5.5.E.5 Center vertically, full cover..... | 149 |
| 6.5.5.E.6 Caption positioning..... | 150 |
| 6.5.5.E.7 Precise positioning..... | 150 |
| 6.5.5.E.8 Caption effects..... | 151 |
| 6.5.5.E.9 Caption with text background..... | 152 |
| 6.5.5.E.10 Old classes to create captions for slideshow (deprecated)..... | 153 |
| 6.5.5.F Groups of banners..... | 154 |
| 6.5.6 Animations..... | 156 |
| 6.5.6.A Basic usage..... | 156 |
| 6.5.6.B List of animations..... | 157 |
| 6.5.6.C Additional options..... | 157 |

| | |
|---|------------|
| 6.5.6.C.1 Animation delay..... | 157 |
| 6.5.6.C.2 Animation duration..... | 158 |
| 6.6 Theme customization..... | 160 |
| 7. Theme settings..... | 161 |
| 7.1 Header..... | 162 |
| 7.1.1 Header structure..... | 162 |
| 7.1.2 Header block positions..... | 164 |
| 7.1.3 Header configuration options..... | 165 |
| 7.2 Category View..... | 169 |
| 7.3 Category View (Grid Mode)..... | 171 |
| 7.4 Category View (List Mode)..... | 175 |
| 7.5 Product Page..... | 176 |
| 7.5.1 Product page structure..... | 176 |
| 7.5.2 Product page block positions..... | 183 |
| 7.5.3 Product page configuration options..... | 184 |
| 7.6 Product Labels..... | 193 |
| 7.7 Footer..... | 195 |
| 7.8 Product sliders..... | 196 |
| 7.9 Default Magento blocks..... | 197 |
| 7.10 Disable Sidebar Blocks (On Home Page)..... | 198 |
| 7.11 Customization Settings..... | 199 |
| 7.12 Data Import..... | 200 |
| 7.12.1 Theme Configuration Import/Export..... | 200 |
| 7.12.2 Import Static Blocks and Pages From Selected Demo..... | 204 |
| 7.13 Integrations..... | 207 |
| 8. Theme Design..... | 208 |
| 8.1 Colors..... | 209 |
| 8.2 Font..... | 210 |
| 8.3 Effects..... | 210 |
| 8.4 Page..... | 210 |
| 8.5 Header..... | 211 |
| 8.6 Header – Top..... | 212 |
| 8.7 Header – Primary..... | 212 |
| 8.8 Main Menu..... | 212 |
| 8.9 Main..... | 212 |
| 8.10 Category View..... | 212 |
| 8.11 Product Page..... | 212 |
| 8.12 Slideshow..... | 213 |
| 8.13 Footer..... | 213 |
| 8.14 Footer – Top..... | 213 |
| 8.15 Footer – Primary..... | 213 |
| 8.16 Footer – Secondary..... | 213 |
| 8.17 Footer – Bottom..... | 214 |
| 9. Theme Layout..... | 215 |
| 9.1 Responsive Layout..... | 216 |
| 9.2 Disable responsive layout..... | 218 |

| | |
|--|------------|
| 10. Zoom..... | 219 |
| 10.1 Zoom configuration..... | 219 |
| 10.1.1 General..... | 219 |
| 10.1.2 Image Size..... | 221 |
| 10.1.3 Thumbnails..... | 222 |
| 10.1.4 Lightbox..... | 223 |
| 11. Slideshow..... | 224 |
| 11.1 Slideshow on home page..... | 224 |
| 11.1.1 Add home page slideshow through the module admin panel..... | 224 |
| 11.1.2 Add home page slideshow through the shortcode..... | 225 |
| 11.2 Slideshow on any page..... | 225 |
| 11.2.1 List of parameters..... | 226 |
| 11.3 Create slides..... | 228 |
| 11.3.1 Basic structure of the static block with slides..... | 229 |
| 11.3.2 Content of a slide..... | 231 |
| 11.3.3 Animated content of a slide..... | 232 |
| 11.3.4 Size of slides and banners..... | 234 |
| 11.3.4.A Slideshow with additional side banners..... | 235 |
| 11.4 Create additional side banners..... | 235 |
| 11.5 Import sample slides and shortcodes..... | 238 |
| 11.5.1 Import static blocks with sample slides..... | 238 |
| 11.5.2 Import page with examples of shortcodes..... | 238 |
| 11.6 Full width slideshow on home page..... | 240 |
| 11.7 Slideshow configuration..... | 241 |
| 11.7.1 General..... | 241 |
| 11.7.2 Additional Banners..... | 244 |
| 11.7.3 Navigation..... | 245 |
| 12. Brands..... | 246 |
| 12.1 Basic information..... | 246 |
| 12.1.1 Brand attribute..... | 246 |
| 12.1.2 Brand images..... | 247 |
| 12.1.3 Brand names and image file names..... | 247 |
| 12.2 Brands configuration..... | 249 |
| 12.2.1 General..... | 249 |
| 12.2.2 Brand on Product Page..... | 254 |
| 12.2.3 Brand List..... | 255 |
| 12.2.4 Brand Slider..... | 256 |
| 13. Menu..... | 259 |
| 13.1 Menu configuration..... | 260 |
| 13.1.1 Top Menu..... | 260 |
| 13.1.2 Mobile Menu..... | 263 |
| 13.1.3 Sidebar Menu..... | 264 |
| 13.1.4 Category Labels..... | 269 |
| 13.2 Drop-down boxes..... | 270 |
| 13.2.1 Drop-down box configuration options..... | 270 |
| 13.2.2 Submenu types..... | 274 |
| 13.2.3 Category blocks..... | 277 |
| 13.2.3.A Category blocks in mobile menu..... | 279 |
| 13.2.4 Sample content for category blocks..... | 281 |
| 13.3 Custom links in the menu..... | 284 |

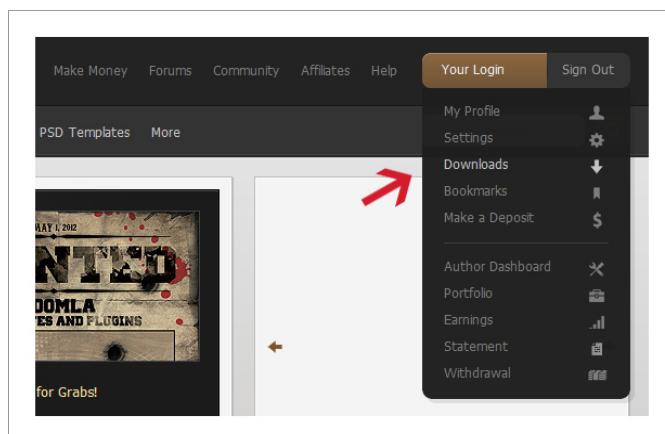
| | |
|---|------------|
| 13.3.1 Custom links inside a static block..... | 285 |
| 13.4 Custom drop-down boxes in the menu..... | 287 |
| 14. Troubleshooting..... | 289 |
| 14.1 General issues after installation..... | 289 |
| 14.1.1 Some features don't work or give errors..... | 289 |
| 14.1.2 Blank page or 404 not found – when trying to open theme admin pages..... | 290 |
| 14.1.3 There is no theme admin tab in the admin panel, or I get a blank page, or Access Denied error..... | 290 |
| 14.1.4 File permissions..... | 290 |
| 14.1.5 Theme features don't work after installation or upgrade..... | 290 |
| 14.1.6 Fatal error: Class 'Mage_Default_Helper_Data' not found..... | 291 |
| 14.1.7 "Package with this name does not exist" message is displayed when trying to enable the theme..... | 291 |
| 14.1.8 After theme installation the store gives blank page or an error..... | 292 |
| 14.1.9 Tabs, sliders or other interactive elements don't work..... | 292 |
| 14.1.10 Images are not displayed..... | 292 |
| 14.1.11 Sliders: previous/next buttons don't work in some browsers..... | 292 |
| 14.1.12 Slideshow is not displayed on the home page..... | 293 |
| 14.2 Configuration..... | 294 |
| 14.2.1 Theme configuration changes do not appear on the frontend..... | 294 |
| 14.2.2 "Failed generating CSS file" message after saving theme configuration..... | 294 |
| 14.3 Third-party extensions..... | 295 |
| 14.3.1 One of my extensions doesn't work with this theme..... | 295 |
| 14.3.2 One of the features stopped working after third-party extension was installed | 295 |

Introduction

Ultimo is a Magento theme with advanced admin module, it's extremely customizable and fully responsive. It is suitable for every type of store. It is a separate design package (not a sub-theme of Magento "default" theme) so you can easily create your own sub-themes.

Rate this theme

Don't forget to rate this theme on [ThemeForest](#) – it's good to know what others think about our work. You can rate it in the download area of your account on ThemeForest:



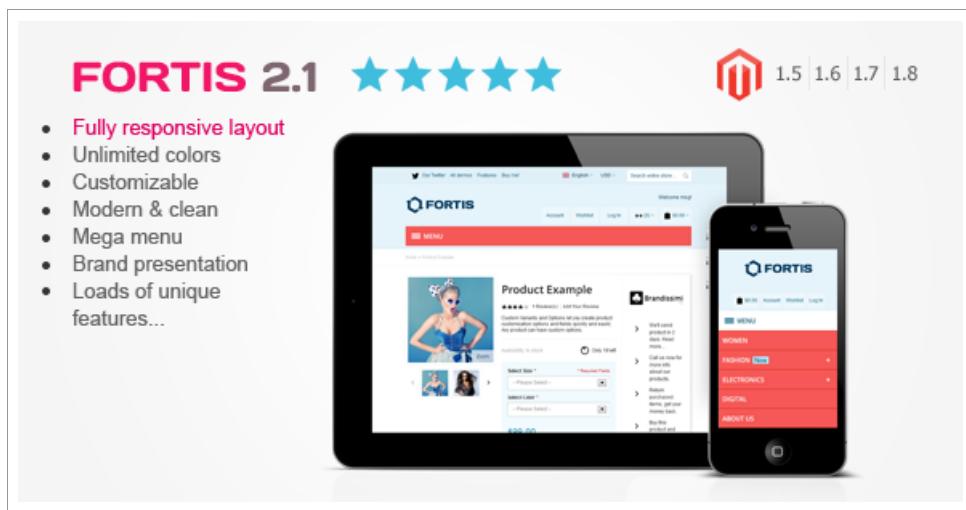
More info

To find out more about theme updates, new features and about our upcoming themes, you can follow us on Twitter and on ThemeForest:



1. FAQ + Quick Start

See also our Fortis theme for Magento (click the image):



1.1 Help & support

1. Theme support

Please read this user guide carefully, it will help you eliminate most of potential problems with incorrect configuration of the theme and Magento. If you don't find the answer to your questions, please refer to this chapter for more information about support policy: [2.2 Support policy](#).

2. Magento support

Magento configuration, installation, maintenance, customization etc. is beyond the scope of our support. But since Magento is very popular platform you can find tons of useful tutorials and articles on the web. Here are some resources for start:

2. Help & Support

1.2 License

With **ThemeForest regular license** you are licensed to use this theme to create one single End Product (the final website customized with your content) for yourself or for one client. In short:

What is allowed with single regular license:

1. You can create one website for yourself or for your client and you can transfer that single website to your client **for any fee**. This license is then transferred to your client.
2. To sell the same website to another client, you will need to purchase **another regular license**.
3. You can install the theme on your **test server** for testing/development purposes (that installation shouldn't be available to the public).
4. You can modify or manipulate the theme, you can **combine it with other works** to create the End Product.
5. Theme **updates** for that single End Product (website) are free.

What is not allowed:

1. With single regular license it is not allowed to create **multiple websites**. For multiple websites you will need multiple regular licenses.
2. **Multi-store Magento installations** are not allowed with single regular license. Each regular license entitles you to use the theme in only one store (store/domain/subdomain). For multiple stores/domains/subdomains you will need multiple regular licenses.

For more information about licenses please refer to ThemeForest:

- License FAQ: <http://themeforest.net/licenses/faq>
- License comparison table: <http://themeforest.net/licenses>
- Regular license details: <http://themeforest.net/licenses/regular>
- Extended license details: <http://themeforest.net/licenses/extended>

1.3 Important things you need to know before you start using Magento

1. [Disable Magento cache](#)

When developing your store you should completely **disable Magento cache**. Enable it after you finish configuring your store: [4.1 Caching system](#).

2. [Reindex Magento](#)

After making changes to your store such as editing products, adding categories etc. you need to reindex the data: [4.2 Index management](#).

1.4 Installation

To install the theme you have two options:

- If you already have a Magento store and want to apply the theme to your store, follow the instructions in [3.2 Installation for live Magento store](#).
- If you want to test the theme and check how the live demo was configured, you can install demo containing the sample data and settings from the live demo. Follow the instructions in [3.1 Installation of demo store](#).

If you encounter any problems after installation, refer to chapter [14. Troubleshooting](#).

1.5 After installation

To start using the theme after installation you need to take a few quick steps to make sure all is configured properly. Also, not all of the features are enabled by default.

1. [Enable the theme in a correct way](#)

Enable the theme in a correct way which is described in the installation instructions: [3.2.2 Install](#).

Important:

do not enable the theme in *System > Design*. It should only be enabled in *System > Configuration > Design*.

In Magento theme can be enabled for the entire store, but you can also set different themes for different parts of the store (even for selected categories, products or CMS pages). If you have already used some other theme in your store, you will need to remove all theme overrides from categories, products and CMS pages. See for more details: [4.3 How to enable the theme](#).

2. Disable cache

In order to be able to see theme configuration changes immediately, you should completely disable the cache. You can enable it after you finish configuring your store. See for more details: [4.1 Caching system](#).

3. Import configuration settings (optional)

After the theme was installed and enabled, you can import theme configuration settings (colors, font, page width etc.) from one of the demo stores. For more details refer to [7.12 Data Import](#). Of course, **configuration import is optional** and you can as well leave the default configuration (by default theme looks the same as demo 1).

Note:

if you decide to import configuration from one of the demo stores, it is recommended to import to the “**Default Config**” scope.

4. Logo

Upload your store logo: [4.4 Logo](#).

5. Home page

Configure your home page: [6.1 Home page and CMS pages](#).

6. Home page slideshow

Configure the home page slideshow: [11. Slideshow](#). Each slide is a static block

which can contain any custom images or HTML. After blocks import, your slideshow will have a few sample slides filled with some dummy images and captions. To edit captions, see this chapter for details: [6.5.5.E.10 Old classes to create captions for slideshow \(deprecated\)](#).

7. Position of “Add to cart” button and product options

Configure position of the **Add to cart** button and product options box on product page: [4.6.1 Position of: Add to Cart button and product options](#).

Note:

you may find these Magento tutorials and documentation useful while configuring your store: [2.1 References](#)

1.6 Fresh Magento installation

If you have just installed a fresh copy of Magento without any sample data, here's where you should start:

1. Disable Magento cache. In order to be able to see configuration changes immediately, you should completely disable the cache. You can enable it after you finish configuring your store. See for more details: [4.1 Caching system](#).
2. Create some categories in *Catalog > Manage Categories* – by default the category tree is completely empty. See for more details: [4.5 Categories](#).
3. Create some products in *Catalog > Manage Products*.
4. Reindex Magento after creating categories and products: [4.2 Index management](#). You will also need to reindex from time to time after making changes in Magento catalog (categories, products, product attributes, attribute sets etc.).

1.7 Theme customization

There are many ways to customize the theme, for more information refer to chapter [5. Customization](#). Customization/modification is beyond the scope of our support.

2. Help & Support



2.1 References

You may find these sites useful while installing and configuring your Magento store:

2.1.1 Magento installation

- **Magento FAQ:**

<http://www.magentocommerce.com/product/faq>

- **System Requirements:**

<http://www.magentocommerce.com/system-requirements/>

- **Server Compatibility Check:**

<http://www.magentocommerce.com/knowledge-base/entry/how-do-i-know-if-my-server-is-compatible-with-magento>

- **Installation Guide:**

<http://www.magentocommerce.com/knowledge-base/entry/magento-installation-guide>

2.1.2 Magento help

Online Magento guide

- Table of contents: [click to see the list of all topics](#)
- Selected topics:
 1. [*Categories, products and attributes*](#)
 2. [*Customers*](#)
 3. [*Payment, shipping and taxes*](#)
 4. [*Promotions and marketing*](#)

Magento Go Knowledge Base

<http://go.magento.com/support/kb>

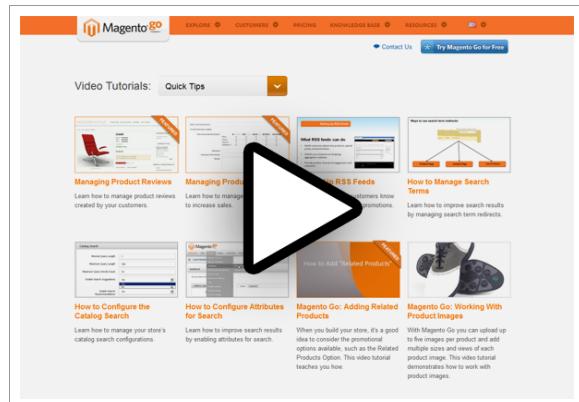


Note:

Magento Go differs from **Magento Community Edition** but most of the functionality is the same so this knowledge base might be very useful.

Magento Go Video Tutorials

<http://go.magento.com/video-tutorials>



Magento Wiki

<http://www.magentocommerce.com/wiki/>

Magento Forum

<http://www.magentocommerce.com/boards/>

stackoverflow.com (Q&A)

<http://stackoverflow.com/questions/tagged/magento>

Magento Stack Exchange (Q&A)

<http://magento.stackexchange.com/>

2.2 Support policy

This user guide was created to help you quickly configure the theme – please read it carefully, it will help you eliminate most of potential problems with incorrect configuration.

Important:

please note that as authors of this theme we are able to provide support only for the features which we created and for the issues related strictly to this theme. We do not provide support for Magento configuration, installation, maintenance, customization etc.

Theme support

If you have found any bugs or have some other problems with this theme, please refer to chapter **14. Troubleshooting**. If the problem is not covered there, you can contact us in the comment section on ThemeForest or through the contact form on our profile page on ThemeForest. The most up-to-date information about support can always be found at:

<http://themeforest.net/item/ultimo-fluid-responsive-magento-theme/3231798/support>

We will respond as soon as possible (within 24 – 48 hours, usually faster). Please make sure you take a look at the available resources before submitting a support request.

Questions about Magento

Elements such as *category management*, *product management*, *checkout process*, *payment methods* and many others are standard Magento functionalities (not related to the theme) and are beyond the scope of our support. How to get Magento support:

1. If you have some questions about Magento itself, please refer to the following websites, almost every detail of Magento configuration was discussed there so you will find answers for most of your questions:
 - **[Magento forum](#)** – <http://www.magentocommerce.com/boards/>
 - **[Magento Stack Exchange](#)** – <http://magento.stackexchange.com/>

- [Stackoverflow.com](http://stackoverflow.com/questions/tagged/magento/) –
<http://stackoverflow.com/questions/tagged/magento/>
2. The best place to start searching is chapter **4. Magento configuration** in this user guide. It covers some of the basic Magento elements.
 3. In the [**Online Magento Guide**](#) you can find description of all the basic Magento settings and many other useful information.
 4. You might also want to check the other sites referenced in **2.1.2 Magento help** such as [**Magento Go Knowledge Base**](#) or [**Magento Go Video Tutorials**](#).

Theme customization/modification support

Customization/modification of Magento themes is a very large subject and can be very complex. Due to that we're unable to provide support related to theme customization. For more information about how you can customize this theme, refer to **5. Customization**.

2.3 Third-party extensions

This theme may not be compatible with some third-party extensions. It's impossible for theme authors to make the theme compatible by default with all existing extensions because there are thousands of available Magento extensions. Only the "default" Magento theme is compatible with all third-party extensions, because all extensions are designed to work with it. But authors of the extension should provide detailed instruction about how to integrate the extension with custom themes, since nobody really uses "default" theme, everybody uses custom themes.

Generally all extensions can work with all themes but they sometimes have to be integrated manually. That's because sometimes the extension and the theme override the same Magento template file so there can be a conflict. In such cases you need to customize the theme and merge two conflicting template files into one template file.

This isn't something specific to our themes, this is general rule for all Magento themes and extensions.

Support

Unfortunately third-party extensions are beyond the scope of our support, please contact the author of the extension if needed. Author of the extension should provide detailed instructions about how to integrate the extension with any custom theme.

How to install

To work with this theme, an extension should be installed in theme's directories, not in the "default" Magento theme directories. To be more specific: extension should be installed in the main directories of the default theme in the Ultimo design package. Here are the main directories of Ultimo:

- Template files:

```
app/design/frontend/ultimo/default
```

- Skin files:

```
skin/frontend/ultimo/default
```

where `ultimo` is the name of the design package, and `default` is the name of the default theme in the package.

In many cases you only need to upload the extension to the theme's directories. Sometimes you need to edit layout files to "tell" the theme where it should display extension's output. If you are not familiar with Magento, you should ask the author of the extension for more information. If you feel that you might have troubles with installing extensions, we advise that you order professional installation services (unfortunately we don't offer this type of services). You can find professional help on sites like, for example, [freelancer.com](https://www.freelancer.com) or [freelanceswitch.com](https://www.freelanceswitch.com).

3. Installation

To install the theme you have two options:

- If you already have a Magento store and want to apply the theme to that store, follow the instructions in [3.2 Installation for live Magento store](#).
- Or if you want to test the theme and check how the live demo was configured, you can install demo containing all the sample data and settings from our live demo. Follow the instructions in [3.1 Installation of demo store](#).

3.1 Installation of demo store

To install demo store with all the sample data (similar to live demo of Ultimo), follow the instruction below. These instructions assume you are in the installation phase with your Magento store. If you already have a working Magento store and database – go to [3.2 Installation for live Magento store](#).

- Navigate inside [Quick Start Demo Package](#), extract the package on your computer and navigate inside.
- Create a new empty database, find [demo_database.sql.gz](#) inside the package and import it into your newly created database (using your database management tool such as *phpMyAdmin* or whatever you use), right before Magento installation. The demo data must be imported before running the Magento Setup Wizard.
- Now you can install Magento: using your FTP client, upload [Magento.zip](#) file to your server (to the folder where Magento can be installed) and extract it there.
- Open a web browser and navigate to your website (to the folder where you extracted [Magento.zip](#)) to load the Magento Setup Wizard.
- Follow the installation instructions step by step just like in case of a standard Magento installation process. During the installation process, enable **Web Server Rewrites** functionality for your Magento (it controls whether Magento will automatically generate search engine friendly URLs for products and categories). It may look like on the following screenshot – option [Use Web Server \(Apache\) Rewrites](#):

The screenshot shows the 'Configuration' step of the Magento 1.x installation process. It includes:

- Database Connection:** MySQL, Database Name: my_database, User Name: my_username, User Password: [redacted], Tables Prefix: [redacted]
- Web access options:**
 - Base URL: http://example.com/
 - Admin Path: admin
 - Additional path added after Base URL: [redacted]
 - Enable Charts: [redacted]
 - Skip Base URL Validation Before the Next Step: [redacted]
 - Use Web Server (Apache) Rewrites: [redacted]
 - Use Secure URLs (SSL): [redacted]
- Session Storage Options:** Save Session Data In: File System

At the bottom right are 'Required Fields' and a 'Continue' button.

- When you're ready to set up your own store, you should install Magento again without the sample data in order to start clean. Using sample data as a base for real store is not recommended.

3.2 Installation for live Magento store

If you already have a Magento store and want to apply the theme, follow the instruction below. To install the theme you only need to upload all the theme files to your server and then enable the theme in the admin panel. But to avoid unexpected problems after installation, you should first prepare your Magento. Please strictly follow the steps described in this guide:

3.2.1 Prepare your Magento

- It is recommended to install the theme on a **test server** before you install it on a live store.

2. Backup Magento files and the store database.

Important:

creating backup before installation of any themes or extensions in Magento is extremely important, especially if you are working on a live store.

3. Disable compilation mode in *System > Tools > Compilation*.

4. Disable all cache systems which you have in your Magento:

- 4.a. Magento's cache (refer to **4.1.2 Disable cache** for details)
- 4.b. full page cache and caching modules for Magento (such as Speedster)
- 4.c. any additional cache on your server, PHP cache engines, APC, etc.

Important:

when developing your Magento store, in order to be able to see changes immediately, you should completely disable the cache. You can enable it after you finish configuring your store.

5. Disable Merge JavaScript Files and Merge CSS Files in *System > Configuration > Developer*. You can enable this feature after installation.

6. Remove all possible custom modifications of the **Magento's "base" theme**. Ultimo (the same as any other Magento theme) relies on "base" theme, so any odifications of the "base" theme can change the default behavior of Ultimo and break some functionality. **You should never edit "base" theme's files**. Custom modifications should be always made via custom sub-themes.

7. Log out from Magento admin panel.

Important:

do not just close the browser window: you need to click the **Log Out** link to refresh the access control system.

8. Before you install, make sure your store is disabled during the installation process. Put Magento in the **maintenance mode**. To do this you can create a simple empty file named `maintenance.flag` in your Magento's root directory. Magento will be offline after that (and you will not be able to log in to the admin panel). You can

remove the file after you finish.

3.2.2 Install

Before you can install the theme you need to prepare your Magento, so please do not omit the previous step: [3.2.1 Prepare your Magento](#).

9. Extract the theme package on your computer and navigate inside [Theme Files](#) directory.
10. Using your FTP client upload [Theme.zip](#) file into the root directory of your Magento installation (the root directory of Magento is the folder that contains files, such as `index.php` and `get.php` and the directories `app`, `js`, `lib`, `skin` and more).
11. Unzip [Theme.zip](#) file – it contains four directories: `app`, `js`, `media`, `skin`.
Magento already contains the same directories - **do not remove those existing directories**. Directories extracted from [Theme.zip](#) have to be merged with existing directories of Magento. Approve, if your system asks you whether you want to merge directories.

Note:

this will not overwrite any core files of Magento, this will only add the theme to your Magento. So if you see a message that those four directories already exist on the server, you can confirm that you want to merge directories.

In case you can't unzip files on your server:

Alternatively, you can unzip [Theme.zip](#) archive on your computer, and upload its content (four directories: `app`, `js`, `media`, `skin`) to your Magento's root directory. But note that files can get broken if you send them via FTP without zipping. So **it is highly recommended to upload zipped files**.

12. **Install theme patches for specific versions of Magento.** In this case patch doesn't fix anything, it only adds theme files which are required in specific versions of Magento. Please note that no patch is needed for Magento 1.7.x.

- 12.a. **NEW:** *Omit this step if you are not using Magento 1.9.3.0 or newer*

This step is required only if you are using Magento **1.9.3.0** (or newer). In this step you will install files which are required only in Magento 1.9.3.0 (or newer). Navigate inside

[Theme \(patch for Magento 1.9.3.x\)](#) directory. Using your FTP client upload content of [Theme \(patch for Magento 1.9.3.x\)](#) directory to

Magento's root directory. Overwrite existing files.

12.b. Omit this step if you are not using Magento 1.9.2.0 or newer

This step is required only if you are using Magento **1.9.2.0** (or newer). In this step you will install files which are required only in Magento 1.9.2.0 (or newer). Navigate inside

Theme (patch for Magento 1.9.2.x) directory. Using your FTP client upload content of Theme (patch for Magento 1.9.2.x) directory to Magento's root directory. Overwrite existing files.

12.c. Omit this step if you are not using Magento 1.9.1.0 or 1.9.1.1

This step is required only if you are using Magento **1.9.1.0** or **1.9.1.1**. In this step you will install files which are required only in Magento 1.9.1.0. Navigate inside Theme (patch for Magento 1.9.1.0, 1.9.1.1) directory. Using your FTP client upload content of Theme (patch for Magento 1.9.1.0, 1.9.1.1) directory to Magento's root directory. Overwrite existing files.

12.d. Omit this step if you are not using Magento 1.8.1.0, 1.9.0.0 or 1.9.0.1

This step is required only if you are using Magento **1.8.1.0**, **1.9.0.0** or **1.9.0.1**. In this step you will install files which are required only in those versions of Magento. Navigate inside

Theme (patch for Magento 1.8.1.0, 1.9.0.0, 1.9.0.1) directory. Using your FTP client upload content of Theme (patch for Magento 1.8.1.0, 1.9.0.0, 1.9.0.1) directory to Magento's root directory. Overwrite existing files.

12.e. Omit this step if you are not using Magento 1.8.0.0

This step is required only if you are using Magento **1.8.0.0**. In this step you will install files which are required only in Magento 1.8.0.0. Navigate inside

Theme (patch for Magento 1.8.0.0) directory. Using your FTP client upload content of Theme (patch for Magento 1.8.0.0) directory to Magento's root directory. Overwrite existing files.

12.f. Omit this step if you are not using Magento 1.6.x

This step is required only if you are using Magento **1.6.x**. In this step you will install files which are required only in Magento 1.6.x. Navigate inside

Theme (patch for Magento 1.6.x) directory. Using your FTP client upload content of Theme (patch for Magento 1.6.x) directory to Magento's root directory. Overwrite existing files.

12.g. Omit this step if you are not using Magento 1.5.x

This step is required only if you are using Magento **1.5.x**. In this step you will install files which are required only in Magento 1.5.x. Navigate inside

`Theme (patch for Magento 1.5.x)` directory. Using your FTP client upload content of `Theme (patch for Magento 1.5.x)` directory to your Magento root directory. Overwrite existing files.

13. Set correct file permissions for all the theme files which you uploaded to your server.

Note:

theme files should have the same permissions as other Magento files. Otherwise Magento will not have access to the theme files, images etc. File permissions may differ depending on server. In case of any doubts about the permissions, please contact your hosting provider. Refer to this article for more details: [Magento Filesystem Permissions](#)

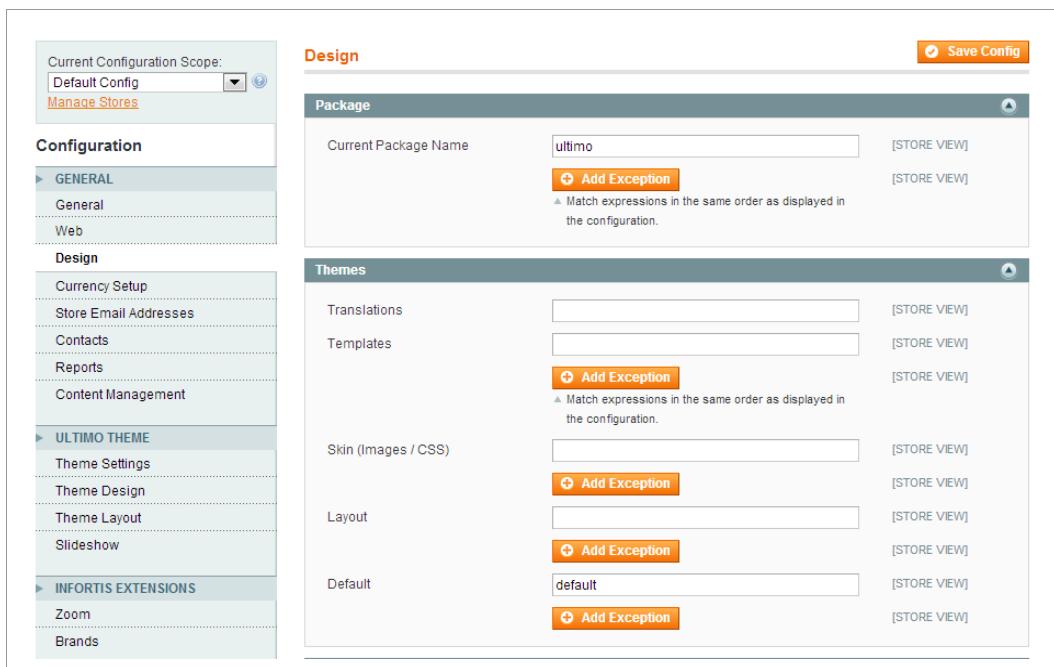
14. At this point, you need to **disable maintenance mode** in your Magento (if you enabled it earlier). To do this you need to remove file `maintenance.flag` which you created inside the root directory of Magento. After that, you will be able to log in to the admin panel again.

Note:

if you get **404 not found** message when trying to open theme configuration pages in the admin panel after installation, you need to **log out** and log in again to refresh the access control system.

15. **Enable the theme.** Log in to the admin panel again. Please note that if you didn't log out before uploading the theme files (like it was described earlier in this chapter), you need to click the **Log Out** link now in order to refresh the access control system. After that you can log in again.

Go to *System > Configuration > Design* section.



In the **Package** tab in the **Current Package Name** field enter the name of the design package: `ultimo`. Next, go to the **Themes** tab and in the **Default** field enter the name of the main theme in the package: `default`. Click **Save Config** button.

Important:

do not enable the theme in *System > Design*. It should only be enabled in *System > Configuration > Design*.

If you have more than one store view in your Magento (by default there's only one) and you want to enable the theme for the entire site, make sure that you select **Default Config** under **Current Configuration Scope** in the top left corner of this section. Otherwise, set the scope according to your needs.

16. Flush Magento cache (refer to **4.1.1 Flush cache**) and any other cache that you have in your Magento installation.
17. Go to the next chapter to import sample blocks and pages.

3.2.3 Import sample blocks and pages

18. You can import sample static blocks and pages provided with the theme. Go to *System > Configuration > Ultimo > Theme Settings > Data Import* to import static blocks and pages. Refer to **7.12 Data Import** for more details about

the import procedure.

Now you can open up your web browser to see the new look of the store. Clear your web browser's cache and cookies for your store domain (this will eliminate potential problems with unrefreshed cache).

Note:

if you get **404 not found** message when trying to open theme configuration pages in the admin panel after installation, you need to **log out** and log in again to refresh the access control system.

All the theme settings in the admin panel can be found under these two tabs:

- *System > Configuration > Ultimo*
- *System > Configuration > Infortis Extensions*

In case of any problems after installation, please refer to the next chapter.

3.3 Problems after installation

Important:

disable Magento cache before you start configuring your store. Otherwise you will need to flush the cache in order to see any changes you have made in the configuration.

If you encounter any problems after installation: flush Magento cache, log out from the admin panel, and log in again. Check again the installation steps and make sure you didn't omit any steps.

What can cause problems:

- a) Magento cache (or external cache) was not flushed after installation of the theme.
- b) CSS files which are responsible for the visual appearance of the theme were not generated automatically. Go to *System > Configuration > Ultimo > Theme Design* and click **Save Config** button. Then go to *System > Configuration >*

Ultimo > Theme Layout and click **Save Config** button again. Then, flush Magento cache and refresh your browser's cache.

- c) You can see all sorts of error messages if the **Compilation** mode was not disabled before installation.
- d) Some of the theme files were not uploaded or are broken. Re-upload original theme files to your server (overwrite existing files) and flush the cache. Note that files can get broken if you send them in large numbers via FTP without zipping. So it is highly recommended to upload zipped files.
- e) File permissions issue – refer to **14.1.4 File permissions** for more details.
- f) Theme wasn't enabled properly. Theme should only be enabled in *System > Configuration > Design* (the same as it was described in the installation steps). Refer to **4.3 How to enable the theme** for more details.
- g) You have some third-party extensions which change/break theme's default behavior or cause some conflicts. Disable all extensions and flush the cache, then enable extensions one by one to check which one is causing problems. Refer to **2.3 Third-party extensions** for more details about third-party extensions.
- h) You edited theme files and accidentally removed or changed something in the files. Re-upload original theme files to your server (overwrite existing files) and flush the cache.
- i) You have customized the theme by creating custom sub-theme and your modifications override or break the default behavior of the theme.
- j) You have some elements that left from your previous theme (e.g. in Magento's "base" theme in `app/design/frontend/base/`) and that breaks the new theme.

Also check your log files in `var/log/` (log settings should be enabled in *System > Config > Developer*) for more information about possible errors.

Please refer to the following chapter: **14. Troubleshooting** for more information about the most common problems.

4. Magento configuration

The following chapter describes some of the basic Magento settings and features.

Please note that as authors of this theme we provide support only for the issues related strictly to the theme. Support for Magento configuration/installation/maintenance etc. is beyond the scope of our support. This chapter should be only treated as a starting point, more information about Magento can be found in Magento documentation: [2.1.2 Magento help](#)

4.1 Caching system

When developing your Magento store, you want to see the changes you have implemented. In order to be able to see changes immediately, you should completely **disable the cache**. You can enable it after you finish configuring your store.

If the cache is enabled, always **refresh the cache** after making any changes in Magento/theme/extension configuration.

To access the cache management screen, go to *System > Cache Management*.

4.1.1 Flush cache

Go to *System > Cache Management*. Click the following buttons to completely clear the store cache:

1. Flush Magento Cache
2. Flush Cache Storage
3. Flush Catalog Images Cache
4. Flush JavaScript/CSS Cache

The screenshot shows the 'Cache Storage Management' section of the Magento Admin Panel. At the top, there are two prominent orange buttons: 'Flush Magento Cache' and 'Flush Cache Storage'. Below these buttons is a table listing various cache types. The table has columns for 'Cache Type', 'Description', 'Associated Tags', and 'Status'. All entries in the 'Status' column are set to 'DISABLED'. The 'Actions' header includes buttons for 'Refresh' and 'Submit'. At the bottom of the screen, there is a section titled 'Additional Cache Management' containing two more orange buttons: 'Flush Catalog Images Cache' and 'Flush JavaScript/CSS Cache'. A small note next to 'Flush Catalog Images Cache' states 'Pregenerated product images files.' and a note next to 'Flush JavaScript/CSS Cache' states 'Themes JavaScript and CSS files combined to one file.'

| Cache Type | Description | Associated Tags | Status |
|----------------------------|--|--------------------------|----------|
| Configuration | System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | DISABLED |
| Layouts | Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | DISABLED |
| Blocks HTML output | Page blocks HTML. | BLOCK_HTML | DISABLED |
| Translations | Translation files. | TRANSLATE | DISABLED |
| Collections Data | Collection data files. | COLLECTION_DATA | DISABLED |
| EAV types and attributes | Entity types declaration cache. | EAV | DISABLED |
| Web Services Configuration | Web Services definition files (api.xml). | CONFIG_API | DISABLED |

4.1.2 Disable cache

Go to **System > Cache Management**. Click **Select All** link, select **Disable** in **Actions** field and click **Submit** button:

| Actions <input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Select Visible"/> <input type="button" value="Unselect Visible"/> 7 items selected | | | | Status <input type="button" value="Disable"/> <input type="button" value="Submit"/> | |
|--|---|--------------------------|--|---|-------------------------------------|
| Cache Type | Description | Associated Tags | Status | | |
| <input checked="" type="checkbox"/> | Configuration System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Layouts Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Blocks HTML output Page blocks HTML. | BLOCK_HTML | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Translations Translation files. | TRANSLATE | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Collections Data Collection data files. | COLLECTION_DATA | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | EAV types and attributes Entity types declaration cache. | EAV | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Web Services Configuration Web Services definition files (api.xml). | CONFIG_API | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |

After cache is disabled cache management screen might look like this:

| Actions <input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Select Visible"/> <input type="button" value="Unselect Visible"/> 0 items selected | | | | Status <input type="button" value="Refresh"/> <input type="button" value="Submit"/> | |
|--|---|--------------------------|--|---|-------------------------------------|
| Cache Type | Description | Associated Tags | Status | | |
| <input type="checkbox"/> | Configuration System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | Layouts Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | Blocks HTML output Page blocks HTML. | BLOCK_HTML | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | Translations Translation files. | TRANSLATE | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | Collections Data Collection data files. | COLLECTION_DATA | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | EAV types and attributes Entity types declaration cache. | EAV | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input type="checkbox"/> | Web Services Configuration Web Services definition files (api.xml). | CONFIG_API | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |

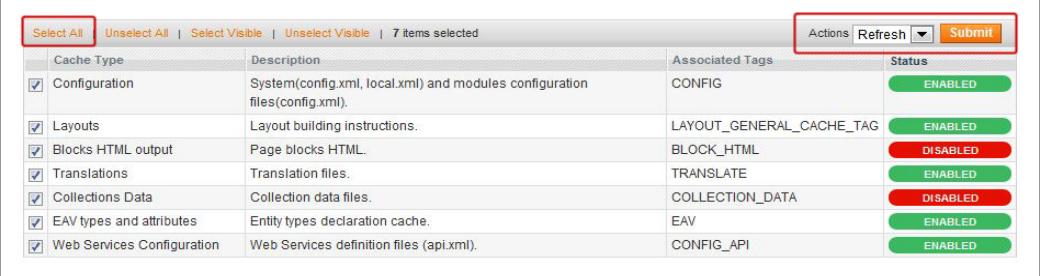
4.1.3 Enable cache

When the store is configured, you can enable caching system again. Go to **System > Cache Management**. Click **Select All** link, select **Enable** in **Actions** field and click **Submit** button:

| Actions <input type="button" value="Select All"/> <input type="button" value="Unselect All"/> <input type="button" value="Select Visible"/> <input type="button" value="Unselect Visible"/> 7 items selected | | | | Status <input type="button" value="Enable"/> <input type="button" value="Submit"/> | |
|--|---|--------------------------|--|--|-------------------------------------|
| Cache Type | Description | Associated Tags | Status | | |
| <input checked="" type="checkbox"/> | Configuration System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Layouts Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Blocks HTML output Page blocks HTML. | BLOCK_HTML | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Translations Translation files. | TRANSLATE | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Collections Data Collection data files. | COLLECTION_DATA | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | EAV types and attributes Entity types declaration cache. | EAV | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |
| <input checked="" type="checkbox"/> | Web Services Configuration Web Services definition files (api.xml). | CONFIG_API | <input type="button" value="ENABLED"/> | <input type="button" value="DISABLED"/> | <input type="button" value="Edit"/> |

4.1.4 Refresh cache

If the cache is enabled, after each configuration change, extension installation, upgrade or uninstallation always refresh the cache. To do this, go to *System > Cache Management*. Click **Select All** link, select **Refresh** in **Actions** field and click **Submit** button:



| Cache Management | | | |
|--|--|--------------------------|-----------------------|
| Cache Type | Description | Associated Tags | Status |
| <input checked="" type="checkbox"/> Configuration | System(config.xml, local.xml) and modules configuration files(config.xml). | CONFIG | ENABLED |
| <input checked="" type="checkbox"/> Layouts | Layout building instructions. | LAYOUT_GENERAL_CACHE_TAG | ENABLED |
| <input checked="" type="checkbox"/> Blocks HTML output | Page blocks HTML. | BLOCK_HTML | DISABLED |
| <input checked="" type="checkbox"/> Translations | Translation files. | TRANSLATE | ENABLED |
| <input checked="" type="checkbox"/> Collections Data | Collection data files. | COLLECTION_DATA | DISABLED |
| <input checked="" type="checkbox"/> EAV types and attributes | Entity types declaration cache. | EAV | ENABLED |
| <input checked="" type="checkbox"/> Web Services Configuration | Web Services definition files (api.xml). | CONFIG_API | ENABLED |

4.2 Index management

Magento indexes most of its data in order to access it faster. If you make changes to your store such as editing products, adding categories, changing urls, etc. you will need to reindex the data so that the changes can show up on your frontend.

To access the index management screen, go to [*System > Index Management*](#). To rebuild your indexes, select all, select **Reindex Data** and click **Submit** button.

Important:

you need to reindex after making any changes in Magento catalog:
categories, products, attributes etc.

4.3 How to enable the theme

In Magento you can enable the theme on many levels. You can set one theme for the entire store, but you can also set different themes for different parts of the store (even for selected categories, products or CMS pages).

If you want all your store pages to look the same (to use the same theme), **your theme should only be enabled** in *System > Configuration > Design* (the same as it was described in the installation steps **3. Installation**). You need to remove all theme overrides from categories, products and CMS pages.

There are many places where a theme can be enabled. If you have just installed a fresh copy of Magento, those places will be empty. Otherwise, please take a look at these sections of the admin panel:

- *System > Design*

Ultimo should not be enabled through this section, delete all existing entries related to Ultimo. By default this section looks like this:

- These fields should be left empty, do not select any theme here:
 - *Catalog > Manage Categories > [category] > Custom Design > Custom Design*
 - *Catalog > Manage Products > [product] > Design > Custom Design*
 - *CMS > Pages > [page] > Design > Custom Theme*

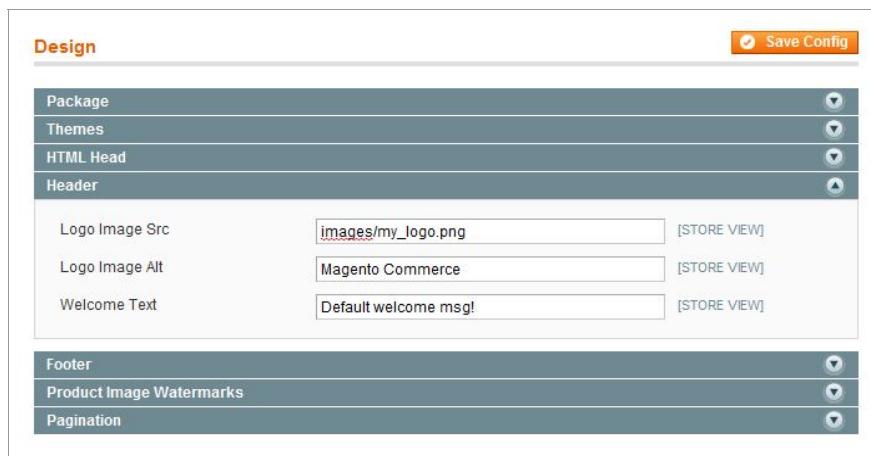
Leave --Please Select-- in these fields, like on the following screenshot:



Remember to **flush the cache** after you change design settings.

4.4 Logo

Upload your logo to skin/frontend/ultimo/default/images/ directory. By default Magento displays logo from the file named **logo.gif**. If you uploaded file with different name or different file format (e.g. *.jpg* or *.png*), go to *System > Configuration > Design > Header* section and in the **Logo Image Src** field specify a path to your logo image. The path is relative to skin/frontend/ultimo/default/ directory.



4.4.1 Sticky logo

You can also display site logo in the sticky header. To display sticky logo:

1. Enable sticky logo under *System > Configuration > Theme Settings > Header*.
2. Upload your sticky logo image to **the same folder where the main logo is stored**.
3. The path of the main logo can be configured under *System > Configuration > Design > Header* in the **Logo Image Src** field.
4. Name your sticky logo file the same as the file with the main logo, but add **_sticky** suffix to the file name.

For example:

if your main logo file is `my-logo.png`, then your sticky logo file has to be named like this: `my-logo_sticky.png`

5. Flush the cache.

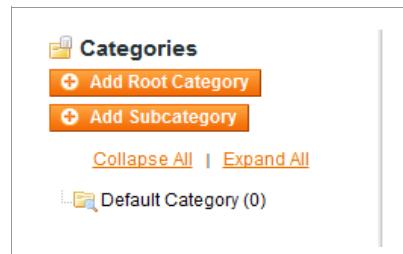
4.5 Categories

Note:

More info about categories in Magento can be found in:

http://www.magentocommerce.com/wiki/welcome_to_the_magento_user_guide/chapter_3#creating_categories

To create and edit categories in Magento, navigate to *Catalog > Manage Categories* section of the admin panel. After installation of Magento your category tree is almost empty, there is only the root category (named **Default Category**):

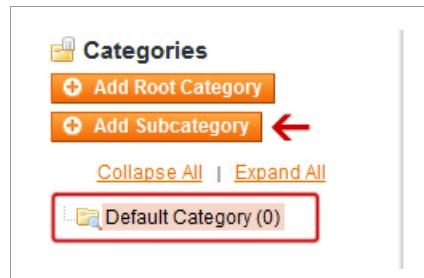


The root category is not displayed in the frontend of the store – **that's why your main menu is empty** just after installing Magento. You need to add some new top-level categories (also called subcategories because they are descendants of the root category).

Important:

all categories have to be **descendants** of the root category. Do not add categories outside of the root category tree.

To add a new category, select the root category by clicking on it and click **Add Subcategory** button:



Fill in all the required fields. Especially set **Is Active** and **Include in Navigation Menu** fields to **Yes**, and click **Save Category**:

A screenshot of a 'New Subcategory' creation form. On the left, there's a sidebar with the 'Categories' menu, the '+ Add Root Category' button, the '+ Add Subcategory' button (which is currently active), and the 'Default Category (0)' link. The main form has a title 'New Subcategory' with 'Reset' and 'Save Category' buttons. It includes tabs for 'General Information', 'Display Settings', 'Custom Design', and 'Category Products'. The 'General Information' tab is selected. It contains several input fields: 'Name *' (with 'Sample category name' placeholder), 'Is Active *' (set to 'Yes'), 'URL Key' (empty), 'Thumbnail Image' (button labeled 'Wybierz plik' with 'Nie wybrano pliku' message), 'Description' (empty), 'Image' (button labeled 'Wybierz plik' with 'Nie wybrano pliku'), 'Page Title' (empty), 'Meta Keywords' (empty), 'Meta Description' (empty), and 'Include in Navigation Menu *' (set to 'Yes').

This will create a new category (which will be displayed as a top-level category in the main menu on frontend):

The screenshot shows the 'Categories' section in the Magento Admin. On the left, there's a tree view of categories: 'Default Category (0)' has a child 'Sample category name (0)'. A red arrow points from the text 'Sample category name (0)' up towards the 'Save Category' button. The right side shows the 'General Information' tab of the 'Sample category name (ID: 3)' edit form. At the top, a green message says 'The category has been saved.' Below it are tabs for 'General Information', 'Display Settings', 'Custom Design', and 'Category Products'. The 'General Information' tab contains fields for Name (Sample category name), Is Active (Yes), URL Key (sample-category-name), Thumbnail Image (Wybierz plik), Description (empty), Image (Wybierz plik), Page Title (empty), Meta Keywords (empty), Meta Description (empty), and Include in Navigation Menu (Yes).

Note that this new category has to be a **descendant of the root category** – it has to be one level lower than the root category:

The image contains two side-by-side screenshots of the Magento Admin Categories page.

Left Screenshot (Correct): Shows the 'Categories' tree. 'Default Category (0)' has children 'Fashion (0)' and 'Books (1)'. A vertical dashed line connects 'Books (1)' to 'Default Category (0)', indicating it is a direct child. The word 'correct' is written in green at the bottom.

Right Screenshot (Not Correct): Shows the same tree structure, but 'Books (1)' is connected by a vertical dashed line to 'Fashion (0)', making it a child of 'Fashion (0)'. The word 'not correct' is written in red at the bottom.

If the category is not displayed in the frontend: **reindex the data** in *System > Index Management* and **flush Magento cache**.

4.5.1 Category info

For each category you can add some basic information which will be displayed on category page (above the products). Navigate to *Catalog > Manage Categories* and select a category:

- **Description** – category description. You can enter text or HTML.
- **Image** – main category image. After image is uploaded and saved, an icon of the existing image will appear to the left of the button, and a **Delete Image** checkbox will appear to the right.

4.5.2 Find category ID

How to find out an internal identifier (ID) of a category through Magento admin panel? Navigate to *Catalog > Manage Categories* and click desired category. Look at the URL of the page in your web browser. It can look like this:

TODO: URL

`http://example.com/admin/catalog/category/edit/id/21/`

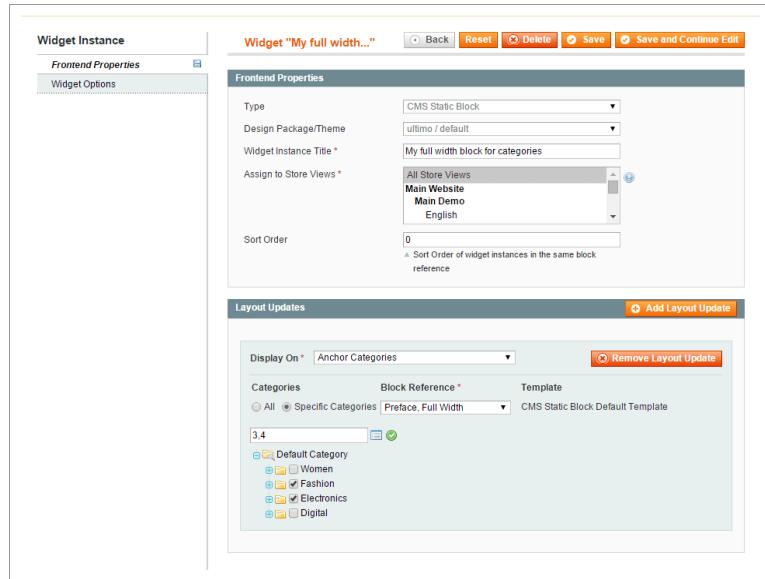
You can find the category ID at the end of the URL. In this example it's **21**.

4.5.3 Landing pages

You can show your customers a landing page instead of the standard product listing page when they select the category. You can do so by enabling static block on your category page (block will be displayed above category products). Navigate to *Catalog > Manage Categories*, select a category, open **Display Settings** tab and select appropriate value in the **Display Mode** field. In the **CMS Block** drop-down, select which block you would like to display for selected category. The list will include all existing static blocks.

There's also another (more flexible) method to add custom static blocks to categories. Go to *CMS > Widgets* and add a new widget instance. Through widgets you can add your static blocks not only to selected categories but also to **selected areas (containers) of the page**. For details refer to chapter **4.8.1 Add blocks to selected pages**.

For example, you can display a full width block (it can contain image banner or any other content) just below the main menu on the category page:



List of all containers available on all pages can be found in [6.2.1 Page](#).

4.5.4 Depth of the menu

In the main menu you can limit number of displayed category levels. To do that, go to [System > Configuration > Catalog](#) and specify the depth in **Maximal Depth** field.

For example, if you want to display only three levels of categories, enter **3** in **Maximal Depth** field and save the configuration. From now on, only three levels of categories will be displayed in the main menu.

4.5.5 Layered Navigation ("Shop By" filters)

Magento's Layered Navigation allows your customers to filter down products using any attribute that is set as filterable for Layered Navigation. The Layered Navigation menu will display in the left column of your category pages, and will contain all of the attributes which are filterable.

You can configure each attribute to be displayed in Layered Navigation: **Use in Layered Navigation** field in [Catalog > Attributes > Manage Attributes > \[select attribute\]](#). Please refer to Magento documentation for more details:

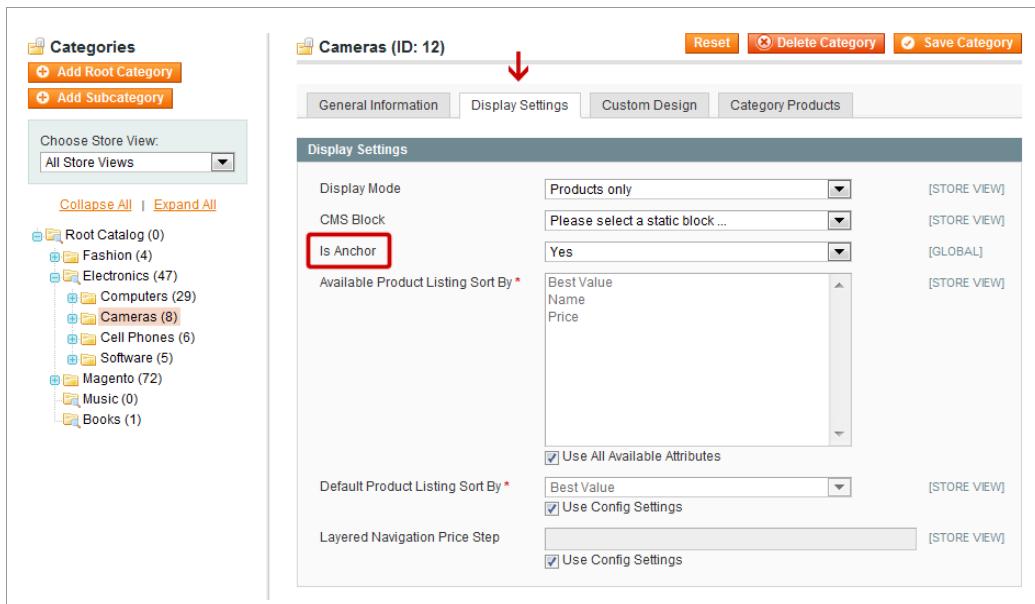
- http://www.magentocommerce.com/wiki/welcome_to_the_magento_user_s_guide/chapter_3#attributes
- http://www.magentocommerce.com/wiki/welcome_to_the_magento_user_s_guide/c

4.5.6 Categories inside Layered Navigation

Note:

This theme adds ability to display a list of categories in the category view (independent from Magento's Layered Navigation). For more details refer to [**13.1.3 Sidebar Menu**](#).

You can display categories **inside** Magento's Layered Navigation block ("Shop by" block) at the top of the left sidebar in category view. To display selected category in the Layered Navigation, go to [*Catalog > Manage Categories*](#), click selected category and set **Is Anchor** field (in **Display Settings** tab) to **Yes**:

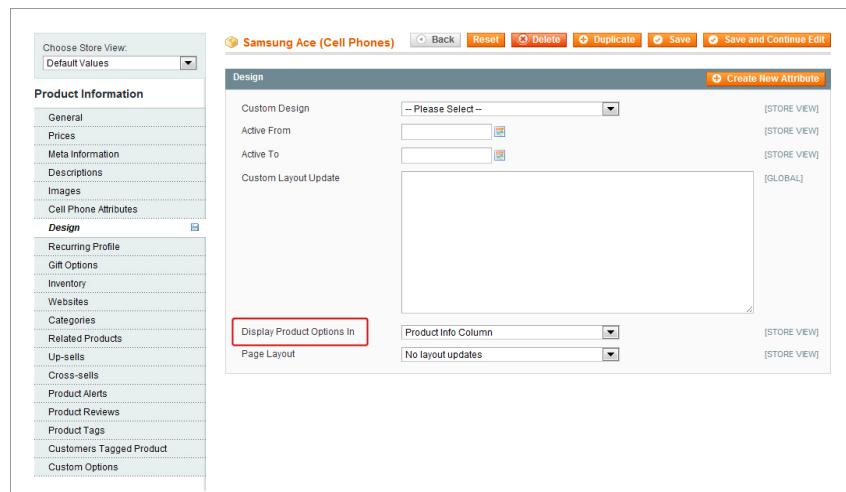


From now on this category will be displayed in the Layered Navigation. If the category is not displayed in the frontend: **reindex the data** in [*System > Index Management*](#) and **refresh Magento cache**.

4.6 Products

4.6.1 Position of: Add to Cart button and product options

For products which have options (like configurable products or bundle products) you can change the position of **options box** and "**Add to cart**" button on product page. These elements can be displayed in the main central column (the column where product name is displayed), or just below the product images (and above tabs). To set the position, select product in *Catalog > Manage Products*, open **Design** tab and select position in **Display Product Options In** field:



The default value of this field ("Block after Info Column") can't be changed anywhere in Magento admin, but instead changing the default value you can simply update that filed in all products at once: use **Update Attributes** action in the **Actions** drop-down list in *Catalog > Manage Products* section.

Important:

to update position of product options and "Add to cart" button on product page in many products at a time, you can use **Update Attributes** action in the **Actions** drop-down list in *Catalog > Manage Products* section.

4.6.2 "Only X left" stock indicator

There is a quantity indicator displayed on the product page. It shows the quantity of items left in stock if the stock level is equal or below the threshold specified in the admin panel. The threshold can be set in **Only X left Threshold** field in *System > Configuration > Inventory*.

For example, if you set the option to 20, all products with quantity equal or less than 20 will have the text **Only X left** on the product page.

For **configurable products** this text is clickable and shows drop-down box with quantities of all products associated with that product.

4.7 Attributes

An attribute in Magento is defined as any property of a product, such as color, price, SKU, manufacturer etc. There are two types of attributes: System Attributes (cannot be deleted and every product must contain them) and Simple Attributes (which can be created by the store admin). Attributes are grouped in Attribute Sets.

Here you can find more info about Magento attributes:

http://www.magentocommerce.com/wiki/welcome_to_the_magento_user_s_guide/chapter_3#attributes

4.7.1 Attribute Management

To edit existing attributes, go to *Catalog > Attributes > Manage Attributes* and select an attribute from the list. To create a new attribute, click **Add New Attribute** button. There are many types of attributes which you can create, one of the most useful is a **drop-down** attribute which lets you define a list of values that will be available when creating new products.

Important:

after making any changes in the attributes and Attribute Sets, you may need to reindex the data in your Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**).

Let's see an example of a **drop-down** attribute `manufacturer` which stores a list of product brands. Fill in the form as follows:

Attribute Information

New Product Attribute

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|----------------------------------|------------------|--------------|---|--|-------|--------|--|--|------------------------------------|----------|--------------|--|----|----------------------------------|-----------------|----|----------------------------------|--|------|---|------------|-------------------|------------------------------------|--|-----|---|---------------------|-----|------------------------|-----|-------------------------|-----|---------------------------|---------------------------|--|--|--|-----|--|--|-------------------------------|----|----------|---|---|--|-----------------------------|----|---|----|-------------------------|----|---------------------------|--|-------------------------------------|----|---------------------------|--|
| Properties | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Manage Label / Options | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <p>Attribute Properties</p> <table border="1"> <tr> <td>Attribute Code *</td> <td>manufacturer</td> </tr> <tr> <td colspan="2">▲ For internal use. Must be unique with no spaces. Maximum length of attribute code must be less than 30 symbols</td> </tr> <tr> <td>Scope</td> <td>Global</td> </tr> <tr> <td colspan="2">▲ Declare attribute value saving scope</td> </tr> <tr> <td>Catalog Input Type for Store Owner</td> <td>Dropdown</td> </tr> <tr> <td colspan="2">Unique Value</td> </tr> <tr> <td>No</td> <td>▲ Not shared with other products</td> </tr> <tr> <td>Values Required</td> <td>No</td> </tr> <tr> <td colspan="2">Input Validation for Store Owner</td> </tr> <tr> <td>None</td> <td>▲</td> </tr> <tr> <td>Apply To *</td> <td>All Product Types</td> </tr> <tr> <td colspan="2">Use To Create Configurable Product</td> </tr> <tr> <td>Yes</td> <td>▲</td> </tr> </table> <p>Frontend Properties</p> <table border="1"> <tr> <td>Use in Quick Search</td> <td>Yes</td> </tr> <tr> <td>Use in Advanced Search</td> <td>Yes</td> </tr> <tr> <td>Comparable on Front-end</td> <td>Yes</td> </tr> <tr> <td>Use In Layered Navigation</td> <td>Filterable (with results)</td> </tr> <tr> <td colspan="2">▲ Can be used only with catalog input type Dropdown, Multiple Select and Price</td> </tr> <tr> <td>Use In Search Results Layered Navigation</td> <td>Yes</td> </tr> <tr> <td colspan="2">▲ Can be used only with catalog input type Dropdown, Multiple Select and Price</td> </tr> <tr> <td>Use for Promo Rule Conditions</td> <td>No</td> </tr> <tr> <td>Position</td> <td>0</td> </tr> <tr> <td colspan="2">▲ Position of attribute in layered navigation block</td> </tr> <tr> <td>Allow HTML Tags on Frontend</td> <td>No</td> </tr> <tr> <td>Visible on Product View Page on Front-end</td> <td>No</td> </tr> <tr> <td>Used in Product Listing</td> <td>No</td> </tr> <tr> <td colspan="2">▲ Depends on design theme</td> </tr> <tr> <td>Used for Sorting in Product Listing</td> <td>No</td> </tr> <tr> <td colspan="2">▲ Depends on design theme</td> </tr> </table> | | Attribute Code * | manufacturer | ▲ For internal use. Must be unique with no spaces. Maximum length of attribute code must be less than 30 symbols | | Scope | Global | ▲ Declare attribute value saving scope | | Catalog Input Type for Store Owner | Dropdown | Unique Value | | No | ▲ Not shared with other products | Values Required | No | Input Validation for Store Owner | | None | ▲ | Apply To * | All Product Types | Use To Create Configurable Product | | Yes | ▲ | Use in Quick Search | Yes | Use in Advanced Search | Yes | Comparable on Front-end | Yes | Use In Layered Navigation | Filterable (with results) | ▲ Can be used only with catalog input type Dropdown, Multiple Select and Price | | Use In Search Results Layered Navigation | Yes | ▲ Can be used only with catalog input type Dropdown, Multiple Select and Price | | Use for Promo Rule Conditions | No | Position | 0 | ▲ Position of attribute in layered navigation block | | Allow HTML Tags on Frontend | No | Visible on Product View Page on Front-end | No | Used in Product Listing | No | ▲ Depends on design theme | | Used for Sorting in Product Listing | No | ▲ Depends on design theme | |
| Attribute Code * | manufacturer | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ For internal use. Must be unique with no spaces. Maximum length of attribute code must be less than 30 symbols | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Scope | Global | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Declare attribute value saving scope | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Catalog Input Type for Store Owner | Dropdown | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Unique Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| No | ▲ Not shared with other products | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Values Required | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Input Validation for Store Owner | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| None | ▲ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Apply To * | All Product Types | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use To Create Configurable Product | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Yes | ▲ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use in Quick Search | Yes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use in Advanced Search | Yes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Comparable on Front-end | Yes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use In Layered Navigation | Filterable (with results) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Can be used only with catalog input type Dropdown, Multiple Select and Price | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use In Search Results Layered Navigation | Yes | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Can be used only with catalog input type Dropdown, Multiple Select and Price | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Use for Promo Rule Conditions | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Position | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Position of attribute in layered navigation block | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Allow HTML Tags on Frontend | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Visible on Product View Page on Front-end | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Used in Product Listing | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Depends on design theme | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Used for Sorting in Product Listing | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ▲ Depends on design theme | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Attribute Information

Edit Product Attribute "Manufacturer"

If you do not specify an option value for a specific store view then the default (Admin) value will be used.

| Manage Titles (Size, Color, etc.) | | | | | | |
|--|---------|------------|------------|----------|-----------------------|--|
| Admin | English | French | German | | | |
| Manufacturer | | Fabriquant | Hersteller | | | |
| Manage Options (values of your attribute) | | | | | | |
| Admin | English | French | German | Position | Is Default | |
| Acer | | | | 0 | <input type="radio"/> | |
| AMD | | | | 0 | <input type="radio"/> | |
| Anashria | | | | 0 | <input type="radio"/> | |
| Apple | | | | 0 | <input type="radio"/> | |
| Argus | | | | 0 | <input type="radio"/> | |
| Asics | | | | 0 | <input type="radio"/> | |
| Canon | | | | 0 | <input type="radio"/> | |
| Crucial | | | | 10 | <input type="radio"/> | |
| New Brand | | | | 20 | <input type="radio"/> | |

After you create the attribute, you need to add it to the Attribute Set which you use for your products. Attribute Sets are described in the next chapter.

4.7.2 Attribute Sets

An Attribute Set is a collection of attributes, created to fit certain types of products. To add a new product in Magento you need to select an Attribute Set which will describe that product. You can create many custom Attribute Sets for different types of products. For example, if you sell t-shirts and books, you could create 2 Attribute Sets: one specific to t-shirts, one specific to books. The Attribute Set for t-shirts can include attributes such as color and size (which are not needed in the Attribute Set for books).

Important:

after making any changes in the attributes and Attribute Sets, you may need to reindex the data in your Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**).

Let's continue an example with `manufacturer` attribute. To add your attribute to an Attribute Set, go to *Catalog > Attributes > Manage Attribute Sets*, select an Attribute Set (default one is named `Default`). You will see **Unassigned Attributes** section with a list of available attributes. You can associate attributes to your Attribute Set by clicking and dragging an attribute into the **Groups** section. To disassociate an attribute, simply drag it back to the **Unassigned Attributes** list.

The screenshot shows the 'Edit Attribute Set 'Default'' interface in the Magento Admin. On the left, there is a 'Edit Set Name' form with a 'Name' field containing 'Default'. A note below says 'For internal use.' In the center, there is a 'Groups' section with a tree view of attributes. Under 'General', attributes like name, sku, weight, status, tax_class_id, url_key, visibility, manufacturer, color, news_from_date, news_to_date, and country_of_manufacture are listed. On the right, there is a 'Unassigned Attributes' list containing a large number of attributes such as activation_information, computer_manufacturers, contrast_ratio, country_origin, cpu_speed, dimension, finish, gender, harddrive_speed, hardrive, in_depth, max_resolution, megapixels, memory, model, processor, ram_size, response_time, room, screensize, and shape. At the top right of the interface, there are 'Back', 'Reset', and 'Save Attribute Set' buttons.

Click **Save Attribute Set** button after you finish. You can add your custom attributes to many Attribute Sets.

Then, go to **Catalog > Manage Products** and open a product. If the product is based on the Attribute Set which contain your `manufacturer` attribute, you will see a list of manufacturers which may look like this:

The screenshot shows the 'Sample Product (Clothes)' edit screen in the Magento Admin. On the left, there's a sidebar with 'Product Information' sections: General, Prices, Clothes, Meta Information, Images, Recurring Profile, Design, Gift Options, Inventory, Websites, Categories, Related Products, Up-sells, Cross-sells, Product Alerts, Product Reviews, Product Tags, Customers Tagged Product, and Custom Options. The 'General' section is selected. On the right, under the 'General' tab, there are several input fields: Name (Sample Product), SKU (top2), Weight (1.0000), and Status (Enabled). There are also dropdowns for URL Key (top2) and Visibility (Catalog, Search). Under 'Country of Manufacture', it says Australia. In the 'Manufacturer' dropdown, 'DummyBrand' is selected, and a list of options is visible: Brandissimi Company, DummyBrand, Manufacturer, Publisher, SampleLogo, WorldCorp, Samsung, and Apple. The 'Manufacturer' dropdown has a blue border around 'DummyBrand', indicating it is the selected value.

If you added `manufacturer` attribute to that Attribute Set, but you don't see the field with the list of manufacturers, you may need to reindex the data in your Magento and flush the cache.

4.8 Content Management System (CMS)

4.8.1 Add blocks to selected pages

Magento Widgets (located under *CMS > Widgets*) allow users to easily add dynamic content to selected pages. This allows for greater control and flexibility in creating informational and marketing content through administrator tools.

Example 1:

Below you can see an example in which we will add a simple block of content to a product page.

To add custom content to selected place, first create a static block. Static blocks in Magento are simple portions of content that can be displayed throughout the site. For more info about creating a static block, refer to next chapter: **4.8.2 Static blocks**.

When your static block with custom content is ready, go to *CMS > Widgets* and add a new widget instance. Widget will add your static block to a selected place on the frontend of your store. Click the **Add New Widget Instance** button. On the next page, you will see the following options:

The screenshot shows the 'Widget Instance' configuration interface. On the left, there's a sidebar with a 'Settings' tab. The main area is titled 'New Widget Instance'. It has a 'Settings' tab selected. Under 'Settings', there are two dropdown menus: 'Type' (set to 'CMS Static Block') and 'Design Package/Theme' (set to 'ultimo / default'). At the bottom right of this section is a blue 'Continue' button. In the top right corner of the main area, there are 'Back' and 'Reset' buttons.

1. Choose the widget type: CMS Static Block.
2. Then you need to select the design in which the widget will be displayed. Select: ultimo/default.
3. Then, click Continue button.
4. On the next page, goto the **Widget Options** tab and click the **Select Block...** button to choose the static block. You will see the list of all your static block. Find and click

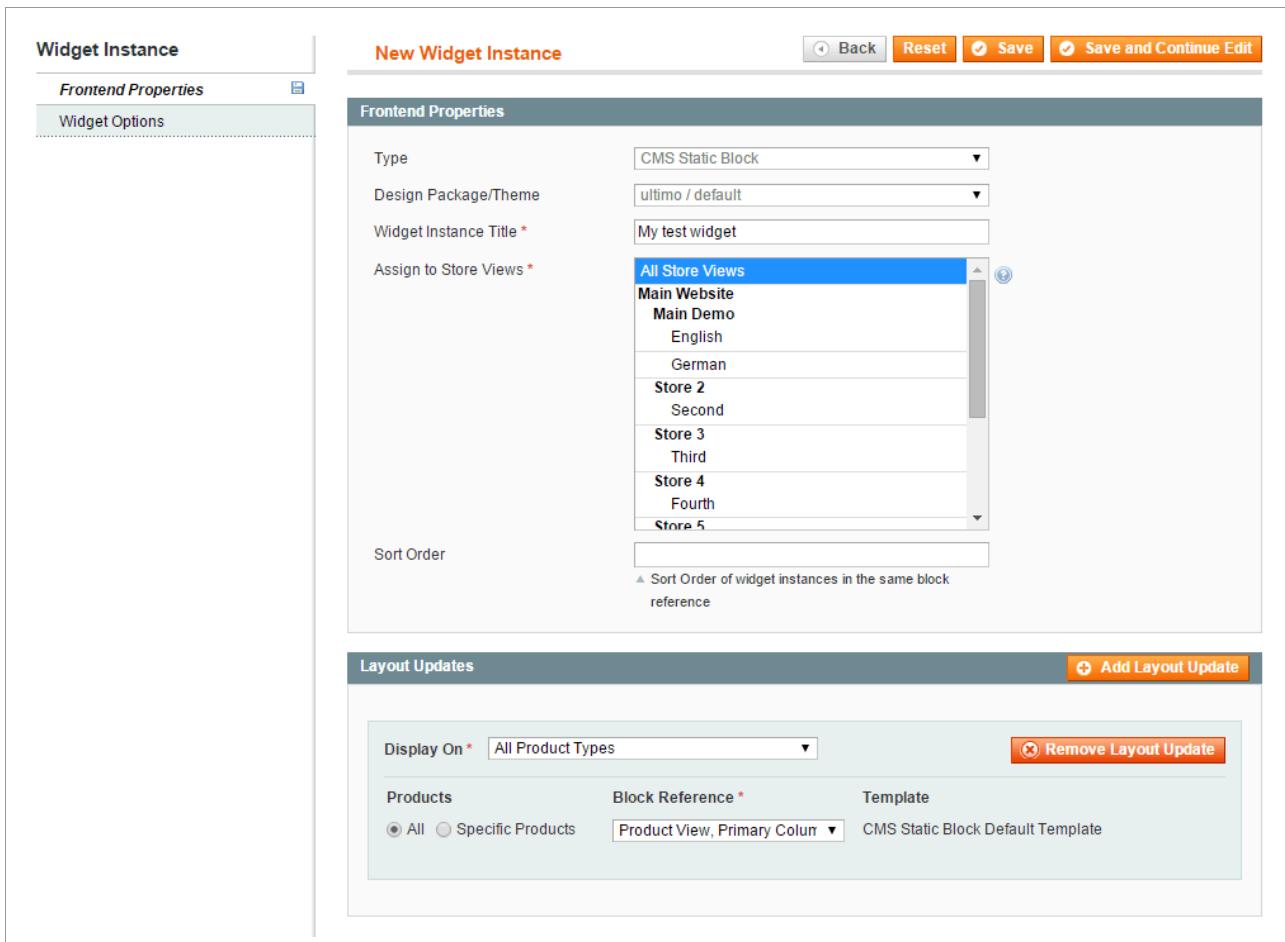
the one which you want to display on the frontend:

The screenshot shows the 'Widget Options' tab of the 'New Widget Instance' interface. At the top right are buttons for 'Back', 'Reset', 'Save', and 'Save and Continue Edit'. On the left, there's a sidebar with 'Widget Instance' and tabs for 'Frontend Properties' and 'Widget Options'. The main area has a title 'Widget Options' and a dropdown menu labeled 'Block *' with 'Not Selected' and a 'Select Block...' button.

5. Next, go to the **Frontend Properties** tab.

The screenshot shows the 'Frontend Properties' tab of the 'New Widget Instance' interface. At the top right are buttons for 'Back', 'Reset', 'Save', and 'Save and Continue Edit'. On the left, there's a sidebar with 'Widget Instance' and tabs for 'Frontend Properties' and 'Widget Options'. The main area has sections for 'Type' (set to 'CMS Static Block'), 'Design Package/Theme' (set to 'ultimo / default'), 'Widget Instance Title' (set to 'My test widget'), and 'Assign to Store Views'. The 'Assign to Store Views' section contains a dropdown menu with several store views listed: 'All Store Views', 'Main Website', 'Main Demo', 'English', 'German', 'Store 2', 'Second', 'Store 3', 'Third', 'Store 4', 'Fourth', and 'Store 5'. Below this is a 'Sort Order' input field and a note about sorting. At the bottom is a 'Layout Updates' section with a 'Add Layout Update' button.

6. In the **Widget Instance Title** field give the widget a title – you can choose any name you like (try to make it informative).
7. If you have a multiple store views, in the **Assign to Store Views** field select in which store views you want to display the widget.
8. The sort order is optional, it will be used to sort widgets if more than one widget was assigned to the same container on the frontend.
9. Click on the **Add Layout Update** button – new options will be displayed inside the **Layout Updates** section:



10. In the **Display On** drop-down select a general type of pages on which the widget will be displayed. Select **All product Types**.

11. Again, new options will be displayed. You will see the contextual options for your selection. In this case, you can select for which products you want to add your widget: **All** or **Specific Products**. Select **All**.

12. In the **Block Reference** field select where exactly on the page you want to display your custom static block. Select one of the product page containers, for example: **Product View, Primary Column, Container 1**.

List of available containers on product page can be found in this chapter **6.2.5 Product page**.

You will notice some additional containers, the list may differ in your specific case – additional containers may have been added by any additional extensions you have installed in your Magento.

13. Finally, save the widget and then flush Magento cache to see the effect on the frontend.

Example 2:

Here's another example in which we will add a block to selected categories (see the **Display On** list). Block will be displayed (see the **Block Reference** field) in the **Preface, Full Width** container, which is a full width container just below the main menu and above the page content area:

The screenshot shows the 'Widget Instance' configuration screen. At the top, there are tabs for 'Frontend Properties' and 'Widget Options'. The 'Frontend Properties' tab is active, showing the following fields:

- Type: CMS Static Block
- Design Package/Theme: ultimo / default
- Widget Instance Title *: My full width block for categories
- Assign to Store Views *: All Store Views (Main Website, Main Demo, English)
- Sort Order: 0

Below this is the 'Layout Updates' section, which includes:

- Display On: Anchor Categories
- Categories: Specific Categories (Preface, Full Width)
- Block Reference: CMS Static Block Default Template
- Template: CMS Static Block Default Template
- Category list: 3,4 (Default Category, Women, Fashion, Electronics, Digital), where 'Fashion' and 'Electronics' are checked.

List of all containers available on each page can be found in [6.2.1 Page](#).

4.8.2 Static blocks

Static blocks in Magento are simple portions of content that can be displayed throughout the site. To manage static blocks, go to **CMS > Static Blocks** section in the admin panel.

To create a static block, go to **CMS > Static Blocks**, click **Add New Block** button and follow these steps:

1. In **Identifier** field enter an identifier of one of the static blocks. Make sure the identifier is lower-case and separated by underscores to follow Magento's standards.
2. Enter the title in **Block Title**.
3. Select the **Store View** to which this block will apply.
4. Select **Enabled** in the **Status** field. **Disabled** means that the block is not displayed in the frontend.
5. Insert your **Content** (it can be text or HTML) and click **Save Block**.

Here's an example:

New Block

General Information

Block Title * Sample title

Identifier * block_header_top_links

Store View * All Store Views
Main Website
Main Store
English
German
French
Store2
Store2 - english

Status * Enabled

Content *

```
<ul>
<li><a href="page1">Sample</a></li>
<li><a href="page2">Header</a></li>
<li><a href="page3">Links</a></li>
</ul>
```

Back Reset Save Block Save and Continue Edit

Note:

when you want to edit static blocks, it's recommended to disable Magento's WYSIWYG so that it doesn't process the content of your block before you start editing it. Turn it on only when it is needed. Go to *Stores > Configuration > Content Management* and set WYSIWYG as **Disabled by Default**.

Important:

in order to see your changes, you need to refresh Magento cache after making any changes in your static blocks.

For more details on creating static blocks, refer to Magento documentation:
http://docs.magento.com/m1/ce/user_guide/content.html

4.8.3 CMS markup tags

There are some special markup tags – a bit of text surrounded by double curly braces – which have a special meaning in Magento. They are also called "CMS tags", "CMS directives", but for simplicity, in this document we will call them "shortcodes".

Shortcodes can be used either in static blocks or CMS pages. For example, rather than hard-coding the full path to a page, you can use a shortcode to represent the store URL. Magento will automatically process all shortcodes during page rendering, so on the frontend instead of shortcodes you will see content generated by each shortcode.

In this chapter you can find examples of the most useful shortcodes available in Magento. For more information, refer to Magento documentation:

http://docs.magento.com/m1/ce/user_guide/cms/markup-tag-syntax.html

4.8.3.A Store URL

The **Store URL** shortcode represents the base URL of your website, and is used as a substitute for the first part of a full URL, including the domain name.

```
 {{store url=""}}
```

This shortcode typed into the content of the page will be replaced with the base URL of your website.

You can use this shortcode to dynamically build all of your links (to avoid hard-coding the domain name). Thanks to that you will not need to rebuild your links after you move your store to a new domain.

For example, if your store domain is `www.example.com`, the following shortcode typed into the page content:

```
 {{store url="path/to/page/about-us"}}
```

will be replaced with this URL:

```
http://www.example.com/path/to/page/about-us/
```

4.8.3.B Media URL

The **Media URL** shortcode represents the location and file name of an image. The shortcode can be used to place an image on a page, static block, or email template.

```
 {{media url="image.png"}}
```

This shortcode typed into the content of the page will be replaced with the URL of the image.

The **image path is relative** to Magento's `media` directory, so you need to store your images inside that directory.

For example, if this is the path of your image: `media/wysiwyg/path/to/image.png`, you can get the URL of that image with the following shortcode:

```
 {{media url="wysiwyg/path/to/image.png"}}
```

Here's how you can use that shortcode in the `src=""` attribute of `` element to display your image on a page:

```

```

4.8.3.C Block ID

The **Block ID** shortcode is one of the easiest to use, and can be used to place a static block directly on a CMS page, or even nested inside another static block. You can use this technique to modify a block for different promotions or languages.

```
{{block id="my-block-id"}}
```

The Block ID shortcode references a static block by its identifier. In the `id` attribute, enter an identifier of the static block which you want to display.

```
{{block id="tester01"}}
```

This shortcode typed into the content of the page will be replaced with the content of the static block `tester01`.

4.8.4 WYSIWYG editor

WYSIWYG (*What You See Is What You Get*) editor is a convenient way to add content to pages and static blocks in Magento.

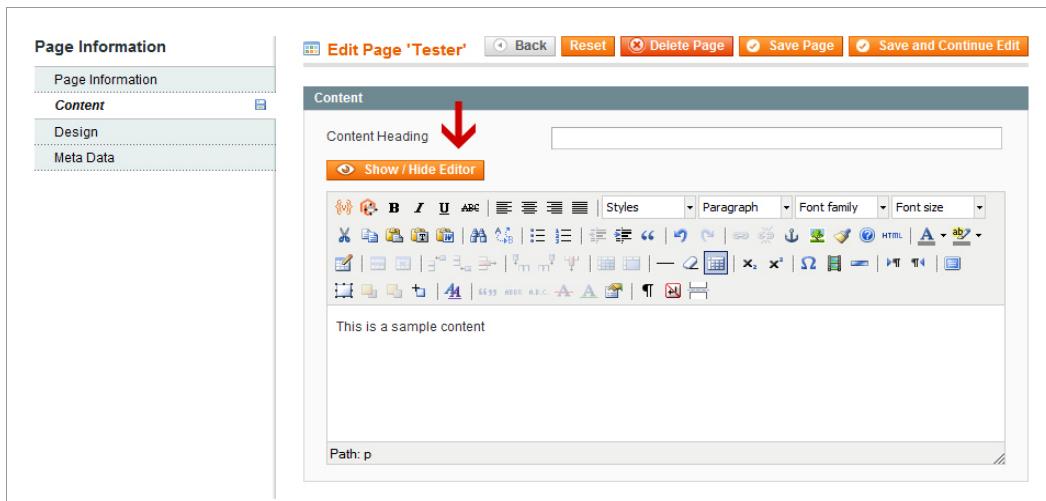
Important:

WYSIWYG should only be used to edit text, **never use it to edit HTML**.

Always turn off WYSIWYG editor when you want to edit HTML or Magento's CMS tags in the page/block content. Otherwise WYSIWYG editor can break your content.

But the best way to use WYSIWYG is to **turn it on only when it is needed**. Go to *System > Configuration > Content Management* and set it as **Disabled by Default**.

After that, if needed, WYSIWYG can always be turned on with **Show/Hide Editor** button above the page/block content field:



4.9 Translation / Localization

4.9.1 Interface translation

This theme introduces a few additional interface character strings. For example “Special Price” label is replaced with shorter “Now only” label. Strings are located in `app/design/frontend/ultimo/default/locale/en_US/translate.csv`. This file can be used to translate the interface into other languages.

Example:

To translate the interface into Spanish follow these instructions:

- Create a new folder for your translation. For Spanish language it will be `app/design/frontend/ultimo/default/locale/es_ES`.
- Copy `translate.csv` from `app/design/frontend/ultimo/default/locale/en_US` and paste it into created folder `app/design/frontend/ultimo/default/locale/es_ES`.
- Open `app/design/frontend/ultimo/default/locale/es_ES/translate.csv` in Open-office Calc (or other text editor like Notepad++, PSPad).

Note:

The editor should be capable to save file in UTF-8 encoding. Do not use Excel, it can break file structure. Read more about useful tools:

http://www.magentocommerce.com/wiki/groups/166/useful_tools_to_work_with_translations

- If you open `translate.csv` in Open-office Calc it might look like this:

| | |
|------------------------|---------------|
| Special Price: | Now only: |
| My Wishlist | Wishlist |
| My Wishlist (%d item) | Wishlist (%d) |
| My Wishlist (%d items) | Wishlist (%d) |

| | |
|------------|---------|
| My Account | Account |
| ... | |

The first column contains the original character strings. It should be **left intact**. In the second column you can place your translation of each string.

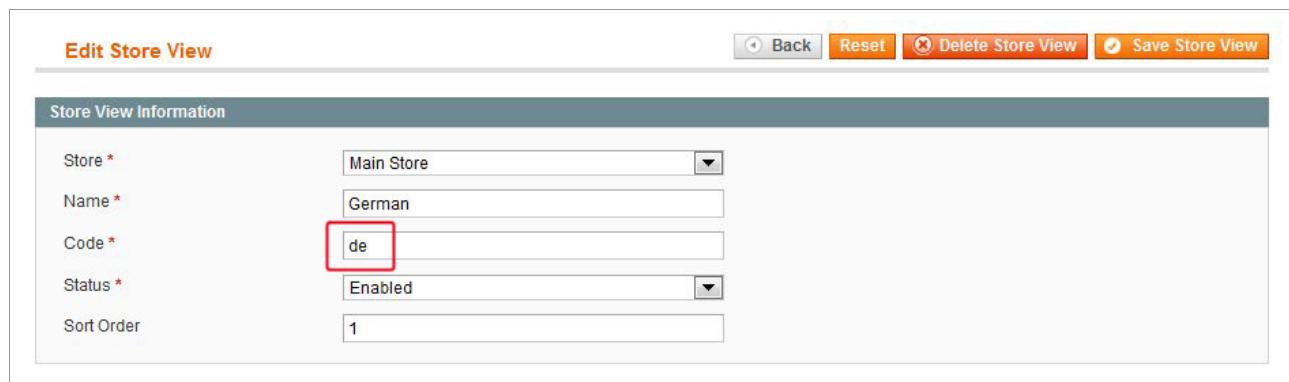
- Some strings contains `%d` or `%s` entries. These entries should be **left intact** in the translated strings.

4.9.2 Language flag

If you have more than one store view in your store, the store view switcher will be displayed at the top of the page. You can enable different language for each store view.

For each available language a flag is displayed in the store view switcher (we can also call it “language switcher”). Flag images (16x12 pixels, PNG format) should be uploaded to `skin/frontend/ultimo/default/images/flags` folder. Image names should be the same as the store view codes. E.g. if you have a store view with the code `de`, you will need to upload a flag image `de.png` to `skin/frontend/ultimo/default/images/flags` folder.

To check what is the store view code, go to `System > Manage Stores` and click on the store view name. You can find the code in the **Code** field:



The screenshot shows the 'Edit Store View' interface. At the top, there are buttons for Back, Reset, Delete Store View, and Save Store View. Below that is a section titled 'Store View Information'. It contains the following fields:

| | |
|------------|------------|
| Store * | Main Store |
| Name * | German |
| Code * | de |
| Status * | Enabled |
| Sort Order | 1 |

Remember that if you change the store view code, you will also need to change the name of the corresponding flag image.

5. Customization

Important:

please note that Magento theme customization is far beyond the scope of this user guide. This chapter is only for informational purposes, you should treat it as a starting point. Before you start to customize the theme you need to be sure that you know what you're doing.

Magento is based on a theming concept referred to as parent/child theming – changes in the theme can be implemented without modifying the original. Magento's "base" theme serves as a parent theme for any custom theme.

The correct way to customize the theme in Magento is to create your own sub-theme of the default theme in the design package. **You should never edit original files of the design package**. Files that need to be modified can be copied from the default theme to your custom sub-theme inside the design package. But don't copy all files, only the files which you're going to modify.

This way you can override any file of the theme. And you can be sure that when something goes wrong you can simply delete your custom sub-theme without breaking the original theme files.

In this chapter you can find the information that will help you to modify the theme.

| | |
|--|-----------|
| 5.1 How to modify the theme..... | 53 |
| 5.1.1 Customize images..... | 53 |
| 5.1.2 Override default CSS styles..... | 54 |
| 5.1.3 Modify template files..... | 55 |
| 5.1.4 Create custom sub-theme..... | 56 |
| 5.2 Magento theme customization..... | 57 |
| 5.2.1 Design packages..... | 57 |
| 5.2.2 Theme structure..... | 58 |
| 5.2.3 Fallback mechanism..... | 59 |
| 5.2.4 Examples of custom sub-theme..... | 61 |
| 5.2.5 Summary..... | 64 |

5.1 How to modify the theme

5.1.1 Customize images

Custom content images

Images which are used in this theme (such as page backgrounds, patterns, banners etc.) are stored in `media/wysiwyg/infortis/` folder. For example:

- `page-background` – images which can be used as page background.
- `slideshow` – images for the home page slideshow.
- `ultimo/_patterns` – this folder contains patterns which can be added to the page background. Patterns can be applied under `System > Configuration > Theme Design`.

To add a custom pattern, replace one of the sample files (for example: `default.custom1.png`) with a custom file and then rename it to `custom1.png` (so you just need to remove the prefix `default.`), like this:

```
default.custom1.png → custom1.png
```

Custom pattern will be visible on the list of patterns in the admin panel.

Interface images

Some interface images are stored in `skin/frontend/ultimo/default/images/` directory. You can replace those files with custom files.

5.1.2 Override default CSS styles

When you want to make some non-standard design changes (for which you can't find any settings in the theme admin panel) you can do it by adding custom CSS styles.

Note:

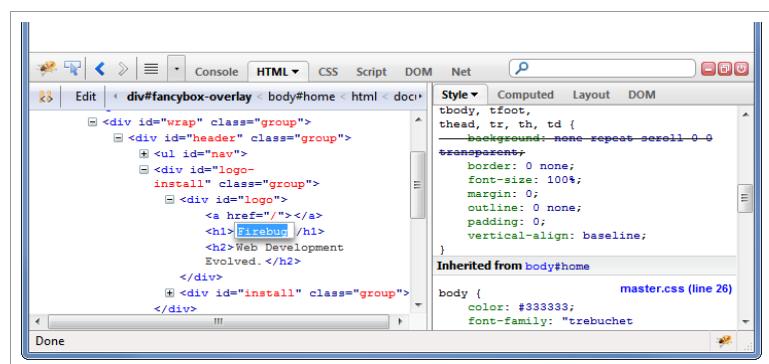
for simple CSS changes there's no need to create custom sub-theme.

In this theme you can enable additional CSS file `custom.css`, in which you can override and extend the default styles of the theme. All the CSS styles added to this file will not be lost after upgrading the theme in the future. Refer to [7.11 Customization Settings](#) to check how to enable that file.

How to find out which part of CSS to override?

How to find out which part of the theme's or Magento's CSS you need to override or extend? The best way is to use **Firebug** (an add-on for Firefox: www.getfirebug.com), **Chrome Developer Tools** (available in Google Chrome) or other similar tools.

See the short video tutorial about **Firebug**: <http://getfirebug.com/video/Intro2FB.mp4>



How to use CSS?

Here you can find more information about CSS:

- <https://developer.mozilla.org/en-US/learn/css>
- <http://www.w3schools.com/css/>
- <http://reference.sitepoint.com/css/syntax>

5.1.3 Modify template files

Magento consists of dozens of template files, each one is used to render a block of content in the frontend of the store. When you want to customize a theme, you can display the name of template file of every block in the frontend by enabling Magento's **Template Path Hints**.

Note:

to enable Magento's **Template Path Hints** go to *System > Configuration > Developer*. Select your current website under **Current Configuration Scope** in the top left corner of the admin panel, otherwise Template Path Hints settings will not be visible.

See this short video tutorial for more details: <http://vimeo.com/1067069>

This way you can easily check which template files you need to edit to customize specific sections of Magento. Files that need to be modified can be copied to your custom sub-theme. Refer to **5.1.4 Create custom sub-theme** for more details about custom sub-themes.

5.1.4 Create custom sub-theme

The correct way to customize the theme in Magento is to create custom sub-theme. You should never edit the original files. Files that need to be modified can be copied to your custom sub-theme. Refer to [5.2 Magento theme customization](#) for more details.

Important:

when you're creating a sub-theme in Magneto you should only copy the files which you want to modify. **Do not copy all files.** Otherwise you will have tons of additional work in the future with any theme upgrade.

Important:

when you're creating a sub-theme, don't copy folder `skin/frontend/ultimo/default/css/_config/` to your sub-theme, that folder should stay in the default theme. Otherwise it will override all your theme configuration from the admin panel.

To check which template file you need to edit to customize specific section of Magento you can enable Magento's **Template Path Hints**. Refer to chapter [5.1.3 Modify template files](#) for more details.

If the file which you selected with **Template Path Hints** doesn't exist in Ultimo package (a theme doesn't need to override all template files from Magento's "base" theme), you can copy that file from Magento's "base" theme to your custom sub-theme inside Ultimo package.

By creating custom sub-theme:

- You can override any file.
- You can create upgrade-proof modifications.
- You can be sure that when something goes wrong you can simply delete your sub-theme without breaking the original files.

5.2 Magento theme customization

5.2.1 Design packages

Themes in Magento are grouped together into design packages (in earlier versions called "interfaces"). By default Magento Community Edition has two packages named "base" and "default".

Note:

do not edit files in the "base" package and do not create custom sub-themes inside the "base" package.

Every design package in Magento comes with a default theme named "default" which is the main theme in the design package. When administrator assigns a package to the store, Magento automatically looks for the theme named "default" in that package.

Design package can also contain variations of the "default" theme. These variations are often called sub-themes because they inherit all the files from the "default" theme (from the same package) and from the "base" package. Inside the sub-theme you can override any file of the "default" theme. Sub-theme can even consist of a single file – other files will be inherited from the "default" theme from the package. This is called a **fallback mechanism** because Magento always falls back to the next theme in the hierarchy to find the requested file. Fallback is described later in this chapter.

5.2.2 Theme structure

Magento themes consists of many files like HTML templates, CSS style sheets, images etc. All files are split between two directories:

- `app/design/frontend/package_name/theme_name/`
Templates directory – contains the layout files, HTML templates and translations
- `skin/frontend/package_name/ theme_name/`
Skin directory – contains CSS style sheets, images and theme-specific JavaScript files

where `package_name` indicates the package name and `theme_name` indicates the theme name.

Template files are organized as follows:

- `layout` – directory contains XML files which define page structure
- `template` – directory contains template files (.phtml), a mix of HTML and PHP
- `locale` – directory contains CSV files with translation strings

Skin files are organized as follows:

- `css` – directory contains CSS files
- `images` – directory contains images
- `js` – directory contains theme-specific JavaScript files

5.2.3 Fallback mechanism

Understanding fallback mechanism is very important when you want to customize the theme, so let's see a simple example.

Let's say we have a design package named "helloworld" with custom sub-theme named "new". Package files are organized as follows:

Default theme in the package:

- app/design/frontend/helloworld/default/
- skin/frontend/helloworld/default/

Your custom sub-theme in the package:

- app/design/frontend/helloworld/new/
- skin/frontend/helloworld/new/

Let's also assume that your custom sub-theme contains a CSS file called "menu.css" located in skin/frontend/helloworld/new/css/menu.css

If your custom theme requests "menu.css" but Magento can't find it in your custom sub-theme in skin/frontend/helloworld/new/css/menu.css, Magento will try to find that file in the next theme in the hierarchy. Next theme in the hierarchy is the "default" theme in "helloworld" package so Magento will search in skin/frontend/helloworld/default/css/menu.css.

If the file is not there, Magento will continue until it locates the file. Next theme in the hierarchy is the "default" theme in "base" package: skin/frontend/base/default/css/menu.css. The "default" theme in the "base" package is the **final fallback point** in the hierarchy.

Let's take a look again at the entire fallback hierarchy:

1. Look for the requested file in custom sub-theme:

skin/frontend/helloworld/new/css/menu.css

2. If not found, look for the file in the "default" theme in the design package:

```
skin/frontend/helloworld/default/css/menu.css
```

3. If not found, look for the file in the "default" theme in "base" design package:

```
skin/frontend/base/default/css/menu.css
```

The first theme in hierarchy is the theme that you assign through the admin panel (in *System > Configuration > Design*). The last theme in hierarchy is the theme "default" in design package "base".

Thanks to the fallback mechanism there is no need to copy all the default theme files to your custom sub-theme when you want to make some changes in your theme. You only copy over the files that you want to edit. Other files will be inherited from the "default" theme in the package and from the "base" package.

The most important advantages of the fallback mechanism:

- selected functionality can be shared between many themes
- less code to maintain
- custom themes are update-proof: you don't edit original files, so your changes are not overwritten after theme updated

5.2.4 Examples of custom sub-theme

Let's see some examples.

Example 1

You have just installed a custom design package named "helloworld" with the default theme (named "default") inside. Let's say that you want to change the font color of the product name on product page.

Package files are organized as follows:

- app/design/frontend/helloworld/default/
- skin/frontend/helloworld/default/

The correct way to customize the "default" theme is to create your own sub-theme in which you will override **selected files** (not all files) of the "default" theme.

1. Let's call the new theme "my_world". Create the following folder inside the "helloworld" package: skin/frontend/helloworld/my_world/. The name of that folder is the name by which Magento will recognize your theme.

2. Create a copy of the CSS file

skin/frontend/helloworld/default/css/styles.css

and paste it into your sub-theme directory:

skin/frontend/helloworld/my_world/css/styles.css

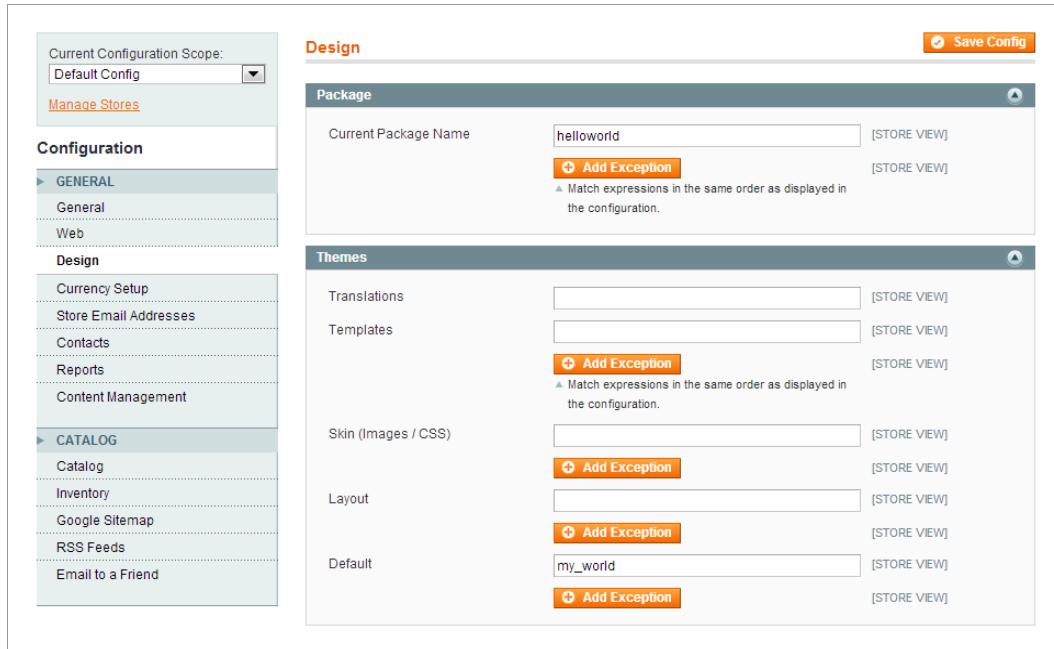
Note that the directory structure inside the sub-theme **must replicate the directory structure of the default theme**:

- skin/frontend/helloworld/default/css/styles.css
- skin/frontend/helloworld/my_world/css/styles.css

3. Open skin/frontend/helloworld/my_world/css/styles.css style sheet and add these lines at the end of the file to override the default color of the product name:

```
.product-view .product-shop .product-name h1 { color:#f00; }
```

4. Enable the new sub-theme in the admin panel. Go to *System > Configuration > Design > Themes* and enter `my_world` (the sub-theme name) in the **default** field:



Finally, refresh Magento cache. From now on your customization will override the default styling of the "helloworld" package.

Example 2

In the previous example we created custom sub-theme named "my_world" in the "helloworld" design package to override one of the skin files of the "default" theme. But you can also override template files located in `app/design/frontend`, not only skin files.

Continuing with the previous example, let's override the contact form template to add simple image banner above the form.

1. Create the following folder for the "my_world" template files in the "helloworld" package: `app/design/frontend/helloworld/my_world/`

The name of the folder has to be the same as the name of the folder which you created for the skin files (`skin/frontend/helloworld/my_world/`)

2. The file which is responsible for the contact form is located here:

```
app/design/frontend/helloworld/default/
template/contacts/form.phtml
```

Create a copy of that file and paste it into your sub-theme directory (remember that the directory structure inside the sub-theme **must replicate the directory structure of the default theme**):

```
app/design/frontend/helloworld/my_world/
template/contacts/form.phtml
```

Note that if the `form.phtml` file doesn't exist in the "helloworld" package, you can copy it from the "base" package (keeping the directory structure intact). "base" package contains all the files that control Magento's default behavior. If the requested file is not found in your custom sub-theme, Magento will try to find that file in the "default" theme from the package. If the file is not found, Magento will call the file from the "base" package.

Thanks to the fallback mechanism, in our example Magento will try to find the file in the following order:

1. `app/design/frontend/helloworld/my_world/
template/contacts/form.phtml`
 2. `app/design/frontend/helloworld/default/
template/contacts/form.phtml`
 3. `app/design/frontend/base/default/
template/contacts/form.phtml`
3. Open `app/design/frontend/helloworld/my_world/template/contacts/
form.phtml` file and insert image banner before the beginning of the contact form.
4. If you didn't enable the new sub-theme in the previous example, you need to do it now. Go to *System > Configuration > Design > Themes* and enter `my_world` (the sub-theme name) in the **default** field. Finally, refresh Magento cache to display changes you have made.

5.2.5 Summary

- Themes in Magento are grouped together into design packages.
- By default Magento comes with the package named "base" which contains all the files that control Magento's default behavior.
- Every design package in Magento comes with a default theme named "default" which is the main theme of the design package.
- Design package can contain variations (sub-themes) of the "default" theme. They inherit all the files from the "default" theme (from the same package) and from the "base" package.
- The correct way to customize the theme in Magento is to create your own sub-theme. You should **never edit original design package files**. Files that need to be changed can be copied to your custom sub-theme.
- This way you can override any file of any design package. And you can be sure that when something goes wrong you can simply delete your custom sub-theme without breaking the original files.
- Magento uses fallback mechanism to make custom themes easier to maintain and upgrade-proof.
- The first theme in the fallback hierarchy is the theme that you assign through the admin panel (in *System > Configuration > Design*). The last theme in hierarchy is the "default" theme in "base" design package.
- If the requested file is not found in your custom sub-theme, Magento will try to find that file in the "default" theme from the package. If the file is not found, Magento will call the file from the "base" package.
- The directory structure inside the sub-theme must replicate the directory structure of the default theme.

6. Theme features and elements

In this chapter you can find description and configuration details of selected elements of this theme, such as home page, built-in static blocks, CSS classes etc.

Note:

The list of all settings available in the theme admin panel in *System > Configuration* can be found in the next chapters.

6.1 Home page and CMS pages

To manage pages in Magento, go to *CMS > Pages* section.

6.1.1 Default pages

To configure the default pages for your store, go to *System > Configuration > General > Web > Default Pages* tab.

The Default Pages configuration determines the landing page that is associated with the base URL, and the corresponding CMS home page. It also determines which CMS page appears when a "404 Page Not Found" error occurs. It also determines if a breadcrumb trail appears at the top of CMS pages.

6.1.1.A Selecting a home page

To configure the default home page for your store, go to *System > Configuration > General > Web > Default Pages* tab, select the desired page in the **CMS Home Page** field and save configuration.

Important:

after the theme is installed, you can import sample versions of home pages from the demo. For more details about import procedure refer to **7.12 Data Import**. Imported pages will be available in the *CMS > Pages* section.

6.1.2 Creating a page

The process of adding a new CMS page to your store is essentially the same for any type of page you might want to create. You can include text, images, blocks of content, variables, and widgets. Most content pages are designed to be read by search engines first, and by people second. Keep the needs of each of these two very different audiences in mind when choosing the page title and URL, composing the meta data, and writing the content.

To create a new page, go to *CMS > Pages* section.

6.1.2.A Page layout

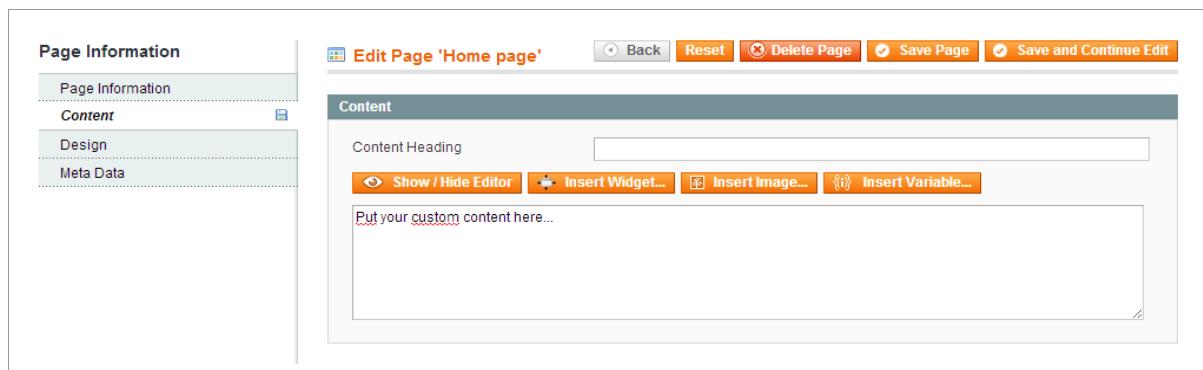
For every CMS page in Magento you can select page layout - the number of columns: one, two, or three columns. To change the layout of the page, go to *CMS > Pages*, select the page, select one of the options in the **Layout** field and click **Save Page** button:

The screenshot shows the 'Edit Page' interface for 'Home page'. On the left, there's a sidebar with tabs: 'Page Information', 'Content' (selected), 'Design', and 'Meta Data'. The main area has two tabs at the top: 'Page Layout' (selected) and 'Custom Design'. In the 'Page Layout' tab, there's a dropdown menu labeled 'Layout*' with '1 column' selected. Below it is a 'Layout Update XML' text area. In the 'Custom Design' tab, there are four dropdown menus: 'Custom Design From', 'Custom Design To', 'Custom Theme' (set to 'Please Select'), and 'Custom Layout' (set to 'Please Select'). There's also a 'Custom Layout Update XML' text area.

6.1.2.B Page content

Page content can include text, images, blocks of content, variables, and widgets.

To edit the content of the page, go to *CMS > Pages* and open the page. Content can be found in the main text field of the **Content** tab:



Important:

always turn off WYSIWYG editor when you want to edit HTML or Magento's CMS tags in the page/block content. Otherwise WYSIWYG editor will break the content. Refer to **4.8.4 WYSIWYG editor** for more details.

6.1.2.C Full width page

To create a full width page, go to *CMS > Pages*, select a page (or create a new one), open the *Design* tab, select *1 column, full width* in the *Layout* field and click the *Save Page* button. From now on, the content of the page will be stretched to full width of the browser viewport.

6.1.2.C.1 Maximum width of the content

If you want the content of that page to keep the same width as a standard page has, you need to wrap the content in an additional div with the `container` class, like on the example below. The content inside this wrapper will be aligned with the other sections of the page (e.g. with the header) and will not exceed the maximum width which is defined in the *Maximum Breakpoint* field under *System > Configuration > Theme Layout*.

Here's the code example:

```
<div class="container">
    <div class="inner-container">
        This is a sample text
    </div>
</div>
```

The `container` class is responsible for the maximum width of the content area. And the `inner-container` class is responsible for additional side padding of the content area.

6.1.2.C.2 Content with full width background

To create full width blocks on the page, wrap your content in an additional div and apply a background color using inline CSS (using a style attribute in HTML elements), for example:

```
<div style="background-color: pink;">
    <div class="container">
        <div class="inner-container">
            This is a sample text 1
        </div>
    </div>
</div>
```

More details about the `background-color` property can be found here:
<https://developer.mozilla.org/en-US/docs/Web/CSS/background-color>

You can add multiple blocks to divide page content into sections, each section in different color:

```
<div style="background-color: pink;">
    <div class="container">
        <div class="inner-container">
            This is a sample text 1
        </div>
    </div>
</div>
<div style="background-color: lightblue;">
    <div class="container">
        <div class="inner-container">
            This is a sample text 2
        </div>
    </div>
</div>
```

6.1.3 Home page blocks

The following chapter describes blocks which can be displayed inside the content area of the home page.

Important:

it's highly recommended to turn off the WYSIWYG editor so that it doesn't process the code before you start editing it. Turn it on only when it is needed. Refer to [4.8.4 WYSIWYG editor](#) for more details.

6.1.3.A Image slideshow

To add a slideshow to the home page, refer to chapter [11. Slideshow](#).

6.1.3.B Banners

To create image banners, refer to chapter [6.5.5 Banners](#).

6.1.3.C Grid/List of Products

To display a block of products (grid or list mode) similar to the block of products displayed in the category view – use the following shortcode:

```
 {{block
    type="catalog/product_list"
    template="catalog/product/list.phtml"
    category_id="4"
    hide_toolbar="1"
}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a block** during page rendering.

The shortcode can take a few parameters. All available parameters are presented below.

Required parameters:

- `category_id` – identifier of the category which you want to present in the block. Make sure you enter identifier of the category which actually exists in your store (and which is active) – otherwise the block will not appear.

Note:

to find out an identifier of a selected category, refer to chapter [4.5.2 Find category ID](#).

Optional parameters:

- `grid_column_count` – determines the number of products displayed in a single row.
If you omit this parameter, the grid will have the number of columns defined in the admin panel in section `System > Configuration > Theme Settings > Category View (Grid Mode) > Number of Columns`.
- `mode` – choose the mode in which the block displays products. Available values:
 - `grid` – grid mode. This is the default mode. If you want to display grid, you can completely omit this parameter.
 - `list` – list mode
- `size` – changes the size of the elements of the grid, such as product name, price.
Available values:
 - `XL` – elements bigger than the standard size
 - `L` – elements a little bit bigger than the standard size
 - `M` – this is the default size. If you want elements to have the default size, you can completely omit this parameter.
 - `S` – elements a little bit smaller than the standard size
 - `XS` – elements much smaller than the standard size
- `hide_toolbar` – enter `1` to hide the toolbar

6.1.4 Featured Products Slider

Featured products slider can show products from any category (only single category is allowed). Slider can be displayed on any CMS page (and inside any static block). To add a slider to a page, use the following shortcode:

```
 {{block
    type="ultimo/product_list_featured"
    template="catalog/product/list_featured_slider.phtml"
    category_id="3"
    block_name="My Products"
}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a slider** during page rendering.

Later in this chapter you can find examples of the shortcodes: **6.1.4.B Examples of shortcodes**.

Note:

make sure that in the `category_id` parameter you specified an identifier of a category which actually exists in your store (and which is active) – otherwise the slider will not appear. Other parameters are described later in this chapter.

Important:

in order to display the slider correctly, you may need to reindex Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**) after making any changes in your products or categories.

6.1.4.A List of parameters

The shortcode which creates a slider can take a few parameters. Only the `category_id`

parameter is required. The other parameters are optional, they can be used to override the global settings of sliders (global settings can be found in [7.8 Product sliders](#)). If you create multiple sliders on a single page (by adding multiple shortcodes), then in each shortcode you can have different values of parameters.

All available parameters are presented below.

Required parameters:

- `category_id` – identifier of the category which you want to present in the slider. Make sure you enter identifier of the category which actually exists in your store (and which is active) – otherwise the slider will not appear.

Note:

to find out an identifier of selected category, refer to chapter [4.5.2 Find category ID](#).

Optional parameters:

- `block_name` – the name of the block displayed as a heading.
- `product_count` – the total number of products presented in the slider.
- `thumbnails` – display products as thumbnails (small image with name and price displayed at the side of the image). Available values:
 - `L` – image size: 200 x 200 pixels
 - `M` – image size: 150 x 150 pixels
 - `S` – image size: 100 x 100 pixels
 - `XS` – image size: 80 x 80 pixels
- `rows` – products can be displayed vertically in multiple rows (one below another). In this parameter you can specify the number of rows of products.
- `breakpoints` – with this parameter you can specify the number of columns (products) in a row visible with a particular browser viewport width. Use the following format (this is the default value, it will be used if this parameter is not specified in the block code):

```
breakpoints="[0, 1], [320, 2], [480, 3], [768, 4], [992, 5], [1200, 6]"
```

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers **[A, B]** number **A** indicates the web browser viewport width, number **B** indicates the number of columns visible if browser viewport width is greater than the width specified in **A**.

In other words: display **B** columns if viewport width is above **A** pixels.

Below you can find some examples:

- **Example 1:**

```
breakpoints="[0, 2], [480, 3], [992, 5]"
```

This value can be translated as: display 2 columns if viewport width is between 0px and 480px, display 3 columns if viewport width is above 480px, display 5 columns if viewport width is above 992px.

- **Example 2:**

```
breakpoints="[0, 1], [480, 2], [768, 1]"
```

This value is useful if you want to place the slider **inside a narrow container**, e.g. inside a sidebar, or inside a narrow column created with the grid system.

This value can be translated as: display 1 column if viewport width is between 0px and 480px, display 2 columns if viewport width is above 480px, display 1 column if viewport width is above 768px.

This means that **if the screen is wider than 768px** only 1 column will be visible. Thanks to that, the slider can be displayed inside narrow containers.

- **Example 3:**

```
breakpoints="[0, 4]"
```

which can be translated as: always display 4 columns.

- **img_width** – enter product image width. If width is not specified, default width will be used (170 pixels).
- **img_height** – enter product image height. To **keep the original aspect ratio** of

product images, specify the width of images using the `img_width` parameter and **omit** the `img_height` parameter. Thanks to that the height of each image will be calculated automatically based on its width.

- `timeout` – to animate the slider automatically, specify the time (in milliseconds) between transitions. Note that 1000 milliseconds = 1 second.
- `move` – number of products that should move on animation. Allowed values:
 - enter `1` to move one product.
 - enter `0` to move all visible products. This is also the default value which is used if the parameter is not specified.
- `loop` – enter `1` if the slider should loop.
- `pagination` – enter `1` to display pagination.
- `centered` – enter `1` to align elements of the slider to the center.
- `size` – with this parameter you can decrease the size of the elements of the slider, such as product name, price. Available values:
 - `XL` – elements bigger than the standard size
 - `L` – elements a little bit bigger than the standard size
 - `M` – this is the default size. If you want elements to have the default size, you can completely omit this parameter.
 - `S` – elements a little bit smaller than the standard size (the old value `size-s` was replaced with `S`).
 - `XS` – elements much smaller than the standard size (the old value `size-xs` was replaced with `XS`).
- `is_random` – enter `1` to display random products from the selected category.
- `sort_by` – with this parameter you can change the order of products. Available values:
 - `position` – default order of products in category. This is the default sort order if you omit this parameter in the shortcode.
 - `name` – sort by name

- `price` – sort by price
- `sort_direction` – with this parameter you can change the direction of sorting. Available values:
 - `ASC` – ascending (default)
 - `DESC` – descending
- `is_responsive` – enter `0` if the slider **should not be responsive**. If this parameter is not specified, it is always assumed that the slider is responsive.

Important:

If the slider **is not responsive**, the number of visible products should be specified with the additional `show_items` parameter (see below).

- `show_items` – the number of products in a row in the non-responsive slider. Use this parameter only if responsive behavior was disabled with parameter `is_responsive` (see above)
- `hide_button` – enter `1` to hide "Add to cart" button
- `hide_labels` – enter `1` to hide product labels (such as "sale")
- `hide_addto_links` – enter `1` to hide "Add to Wishlist" and "Add to Compare" links
- `hide_name` – enter `1` to hide product name
- `hide_rating` – enter `1` to hide product ratings
- `hide_price` – enter `1` to hide product price

6.1.4.B Examples of shortcodes

Sort by name, automatic scrolling

To change the default sort order, use `sort_by` and `sort_direction` parameters.

To add the automatic scrolling, use the `timeout` parameter. For example, `timeout="8000"` will animate the slider every 8 seconds (8000 milliseconds equals 8 seconds).

```
 {{block
type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="My Products"
sort_by="name"
sort_direction="ASC"
timeout="8000"
hide_button="1"
}}}
```

Large products with large images

To make the products larger, you need to change the value of the `breakpoints` parameter to change the number of products in a row.

Also, you need to increase the size of product images using `img_width` and `img_height` parameters. To **keep the original aspect ratio** of images, specify only the width and **omit** the `img_height` parameter. Thanks to that the height of each image will be calculated automatically based on its width.

```
 {{block
type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Our Featured Products"
breakpoints="[0, 1], [480, 2], [640, 3], [992, 4]"
img_width="265"
centered="1"
hide_button="1"
}}}
```

Vertical thumbnails

To display products as thumbnails, add the `thumbnails` parameter to specify the size of the thumbnails.

To display products vertically in multiple rows (one below another), specify number of rows using the `rows` parameter.

Also, if you want to place the slider with vertically displayed thumbnails inside a narrow column (e.g. inside a sidebar, or inside a narrow column created with the grid system), it's

recommended to set the `breakpoints` parameter to `[0, 1], [480, 2], [768, 1]`, so that the slider displays only one column of products when screen is wide (on large tablets and desktops).

```
 {{block
type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Vertical Thumbnails"
thumbnails="M"
breakpoints="[0, 1], [480, 2], [768, 1]"
rows="4"
hide_button="1"
}}}
```

Small vertical thumbnails

This example is similar to previous examples. What's different is that you can additionally make the thumbnails smaller by changing the value of the `thumbnails` parameter to `S`, and make the elements of products (name, price) smaller by adding `size="S"` parameter.

You can also hide elements such as: "Add to cart" buttons, product labels, "Add to..." links, ratings etc.

```
 {{block
type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml"
category_id="3"
product_count="12"
block_name="Small Thumbnails"
thumbnails="S"
breakpoints="[0, 1], [480, 2], [768, 1]"
rows="4"
size="S"
hide_button="1"
hide_labels="1"
hide_addto_links="1"
hide_rating="1"
}}}
```

6.1.4.C Display any set of products using the slider

You can use the slider to display any set of products in your store. Slider can display products **from any category**, so you can simply create categories which will group any type of products you need, for example: bestsellers, products from specific brand, new products, promotional products etc.

You can, for example, display "on sale" products on the home page:

1. Create a new category. Name it "Special Offer".
2. Set the *Include in Navigation Menu* field to **No** – thanks to that the category will not be displayed in the main menu. The category will be only used to group selected products for slider.
3. Add all products with special price (or any other products you want) to the new "Special Offer" category.
4. Make sure Magento **indexes** are refreshed after you added products to the new category.
5. Create a slider (using a shortcode) which will display products from the "Special Offer" category:
 - a) Create a shortcode which will define the slider. Examples of shortcodes and description of all available parameters can be found earlier in this chapter.
 - b) Specify the identifier of the "Special Offer" category in the shortcode. Make sure you use an identifier of a category which really exists in your store – otherwise the slider won't appear.

Note:

to find out an identifier of selected category, refer to chapter **4.5.2 Find category ID**.

- c) Add the shortcode to the content of your page.
6. Flush Magento cache (and any other cache you use).

6.1.5 New Products Slider

New Products Slider can display products which are marked as new in your store. To mark the product as “new”, go to *Catalog > Manage Products*, select the product and in **Set Product as New from Date** and **Set Product as New to Date** fields, enter the date range in which the product will be promoted as a new product.

To display the slider, enter the following code in the content field of any page or static block, Magento will automatically replace the code with a slider during page rendering:

```
{block type="catalog/product_new" template="catalog/product/new.phtml" products_count="10"
hide_button="1" block_name="My New Products"}
```

Important:

In order to display this block properly, you may need to reindex Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**) after marking products as new and after making changes to the catalog (adding or editing categories, products, attributes etc.).

Parameters

Global settings related to product sliders can be found in **7.8 Product sliders**. Global settings can be overridden per slider with the following parameters:

Required parameters:

- **products_count** – the total number of products presented in the slider. Note the letter “s” at the end of “products”.
- **block_name** – the name of the block displayed as a heading.

Optional parameters:

- **breakpoints** – with this parameter you can specify the number of products in a row

visible with a particular browser viewport width. Use the following format (this is the default value, it will be used if this parameter is not specified in the block code):

```
breakpoints="[0, 1], [320, 2], [480, 3], [768, 4], [992, 5], [1200, 6]"
```

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of products visible if browser viewport width is greater than the width specified in `A`.

In other words: display `B` products if viewport width is above `A` pixels. Below you can find some examples:

- **Example 1:**

```
breakpoints="[0, 2], [480, 3], [992, 5]"
```

which can be translated as: display 2 products if viewport width is between 0px and 480px, display 3 products if viewport width is above 480px, display 5 products if viewport width is above 992px.

- **Example 2:**

```
breakpoints="[0, 4]"
```

which can be translated as: always display 4 products.

- **Example 3:**

```
breakpoints="[0, 3], [992, 6]"
```

which can be translated as: display 3 products on narrow screens (if viewport width is below 992px), and display 6 products on wider screens (above 992px).

- **is_responsive** – enter `0` if the slider **should not be responsive**. If this parameter is not specified, it is always assumed that the slider is responsive.

Important:

If the slider **is not** responsive, the number of visible products should be specified with the additional **show_items** parameter (see below).

- **show_items** – the number of products in a row in the non-responsive slider. Use this parameter only if responsive behavior was disabled with parameter **is_responsive** (see above)
- **timeout** – to animate the slider automatically, specify the time (in milliseconds) between transitions. Note that 1000 milliseconds = 1 second.
- **move** – number of products that should move on animation. Allowed values:
 - enter **1** to move one product.
 - enter **0** to move all visible products. This is also the default value which is used if the parameter is not specified.
- **loop** – enter **1** if the slider should loop.
- **pagination** – enter **1** to display pagination.
- **img_width** – enter product image width.
- **img_height** – enter product image height. Do not specify this parameter if you want to keep the aspect ratio of the product images – if height is not specified, it will be calculated automatically based on width.
- **centered** – enter **1** to align elements of the slider to the center.
- **size** – with this parameter you can decrease the size of the elements of the slider, such as product name, button, product labels. Available values:
 - **size-s** – elements a little bit smaller than the standard size
 - **size-xs** – elements much smaller than the standard size
- **hide_button** – enter **1** to hide “Add to cart” button (by default displayed below each product).
- **init_delay** – this parameter is no longer supported.

Examples

Here you can see some examples of the block code:

```
 {{block type="catalog/product_new" template="catalog/product/new.phtml" products_count="10"
centered="1" size="size-s" hide_button="1" block_name="New Products"}}
```

```
 {{block type="catalog/product_new" template="catalog/product/new.phtml" products_count="20"
breakpoints="[0, 1], [320, 2], [480, 3], [768, 4], [992, 5], [1200, 6]" hide_button="1"
block_name="New products in our store"}}
```

6.2 Built-in CMS blocks

There are more than 40 predefined CMS blocks (static blocks) – content placeholders displayed in almost every part of the store, ready to show any content you need: text or HTML. You can import all of these blocks with one click – refer to [7.12 Data Import](#) for more details about the import.

Important:

content of all static blocks is also available in XML format in case you need to restore the original content of selected blocks. XML can be found in the following directory:

```
app/code/local/Infortis/Ultimo/etc/import/
```

Important:

always turn off WYSIWYG editor when you want to edit HTML or Magento's CMS tags in the page/block content. Otherwise WYSIWYG editor will break the content. Refer to [4.8.4 WYSIWYG editor](#) for more details.

After importing, some of the blocks are disabled. You will need to enable them in order to show their content in your store. You can replace the default content of each block with your own content.

Note:

in order to see your changes, you need to refresh Magento cache after making any changes in your static blocks.

For more information about static block management in Magento refer to chapter **4.8.2 Static blocks**.

Identifiers of all built-in blocks are described later in this chapter. For example, `block_footer_payment` is an identifier of the block which displays payment information in the footer section. So if you enable that static block, the content of that block will be automatically displayed in the footer of your store.

Sample content

Most of the blocks already contain sample content. You can replace it with any custom content, but using provided samples will help you to keep the design consistent. For example, in all blocks containing links you can use the following HTML template:

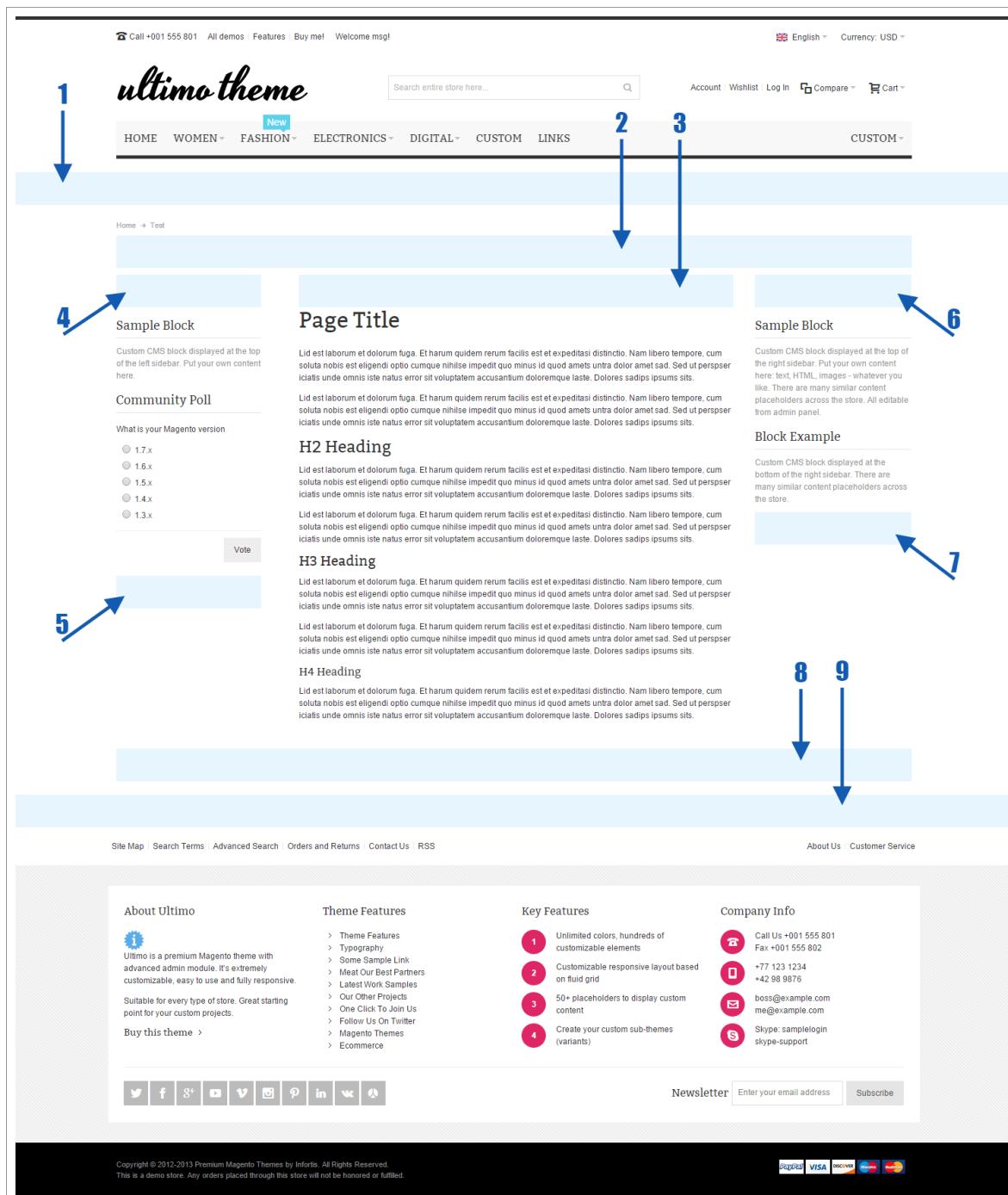
```
<ul class="links">
    <li class="first">
        <a href="{{store url="path/to/page"}}>First Sample Link</a>
    </li>
    <li>
        <a href="{{store url="path/to/page"}}>Second Sample Link</a>
    </li>
    <li class="last">
        <a href="{{store url="path/to/page"}}>Last Sample Link</a>
    </li>
</ul>
```

Replace `path/to/page` with your own value. It will be appended to the store's base URL with additional slash (/) at the end of the resulting URL.

Below you can find a list of identifiers of all built-in static blocks available in this theme.

6.2.1 Page

The screenshot below presents positions of block containers which are available on every page of the store, including home page, product pages, category pages.



You can assign any number of custom **static blocks** to each container. If block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter [4.8.1 Add blocks to selected pages](#).

Containers are suitable for any kind of additional information: discounts and promotions, image banners, social services bookmarks etc.

List of containers:

1. **Preface, Full Width** – container above the content area and above sidebars. It is stretched to full width.
2. **Preface** – container above the content area and above sidebars.
3. **Main Content Area** – container above the content area of the page. It is as wide as the content area. It is displayed between the sidebars (if page has sidebars).
4. **Left Column, Top** – container at the top of the left sidebar. Blocks added to this container will be visible only if page has the left sidebar.
5. **Left Column** – container at the bottom of the left sidebar. Blocks added to this container will be visible only if page has the left sidebar.
6. **Right Column, Top** – container at the top of the right sidebar. Blocks added to this container will be visible only if page has the right sidebar.
7. **Right Column** – container at the bottom of the right sidebar. Blocks added to this container will be visible only if page has the right sidebar.
8. **Postscript** – container below the content area and below sidebars.
9. **Postscript, Full Width** – container below the content area and below sidebars. It is stretched to full width.

6.2.2 Header

The screenshot below presents positions of block containers which are available in the page header:



You can assign any number of custom **static blocks** to each container. If block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter [4.8.1 Add blocks to selected pages](#).

List of containers which are available in the page header:

1. **Page Header, Top, Left** – container at the left side of the top header. Blocks added to this container are floating from left to right.

Note:

for backward compatibility, static blocks with ID `block_header_top_left` and `block_header_top_left2` (if they exist in your Magento) are displayed automatically just next to this container. But these blocks are deprecated and will be removed in next versions.

2. **Page Header, Top, Right** – container at the right side of the top header. Blocks added to this container are floating from right to left.

Note:

for backward compatibility, static blocks with ID `block_header_top_right` and `block_header_top_right2` (if they

exist in your Magento) are displayed automatically just next to this container. But these blocks are deprecated and will be removed in next versions.

3. **Page Header, Primary, Left Column** – container displayed above the content of the left column in the primary header.
4. **Page Header, Primary, Central Column** – container displayed above the content of the central column in the primary header.
5. **Page Header, Primary, Right Column** – container displayed above the content of the right column in the primary header.
6. **Page Header** – container displayed above the columns of the primary header.
7. **Page Top** – full-width container displayed above the header.

Additionally you can also display two predefined static blocks in the menu bar:

8. **block_nav_links** – to display custom links in the main menu bar, create a static block with identifier **block_nav_links**. The content of this block will be displayed directly in the menu bar so the content needs to have proper structure (it's described in this chapter: [13.3.1 Custom links inside a static block](#)).

There is another method to add custom links to the menu, for more details refer to [13.3 Custom links in the menu](#).

9. **block_header_nav_dropdown** – to display a custom drop-down box in the main menu, create a static block with identifier **block_header_nav_dropdown**. The content of this block will be displayed in a drop-down box and the title of this block will be displayed as the name of the menu item.

This block is deprecated and will be removed in future versions. There is a better method to add custom drop-down boxes to the menu, for more details refer to [13.4 Custom drop-down boxes in the menu](#).

There's also an additional container displayed only in mobile view:

10. **Page Mobile Header, Top – container** in the top header. Custom static blocks assigned to this container via Magento widgets will be automatically displayed in the top header only in mobile view.

For quick start you can use sample static block `sample_mobile_header_top` which is provided with the theme (you just need to import static blocks first). Simply create a new widget and inside that widget assign the static block `sample_mobile_header_top` to the container named `Page Mobile Header, Top`, like this:

The screenshot shows the 'Widget Instance' configuration screen. The title bar says 'Widget "Sample - mobile header - top"'. The main area is divided into 'Frontend Properties' and 'Layout Updates' sections.

Frontend Properties:

- Type: CMS Static Block
- Design Package/Theme: ultimo / default
- Widget Instance Title: Sample - mobile header - top
- Assign to Store Views: All Store Views (Main Website, Store 1, Demo1, Store 2, Demo2, Store 3, Demo3, Store 4, Demo4, Store 5, Demo5)
- Sort Order: 0

▲ Sort Order of widget instances in the same block reference

Layout Updates:

- Display On: All Pages
- Block Reference: Page Mobile Header, Top
- Template: CMS Static Block Default Template

Buttons at the top right include Back, Reset, Delete, Save, and Save and Continue Edit.

For details about assigning static blocks to containers, refer to chapter [4.8.1 Add blocks to selected pages](#).

You can add any content inside your static block. Below you can find some examples.

Example 1:

A short text with a background color:

```
<div style="background-color: #e4fbf6; padding: 10px;">  
    Text example  
</div>
```

To align the content to the left side, put your content inside a `<div>` with the following classes:

```
<div class="item item-left">  
...  
</div>
```

So the content of your block can look like this:

```
<div class="item item-left">  
    <div style="background-color: #e4fbf6; padding: 10px;">  
        Text example  
    </div>  
</div>
```

To align the content to the right side, replace the `item-left` class with the `item-right` class, like this:

```
<div class="item item-right">  
    <div style="background-color: #e4fbf6; padding: 10px;">  
        Text example  
    </div>  
</div>
```

To make the content centered, remove the `item-left` and `item-right` classes and use an inline CSS `style="text-align: center;"` on the containing `<div>`:

```
<div class="item" style="text-align: center;">  
    <div style="background-color: #e4fbf6; padding: 10px;">  
        Text example  
    </div>  
</div>
```

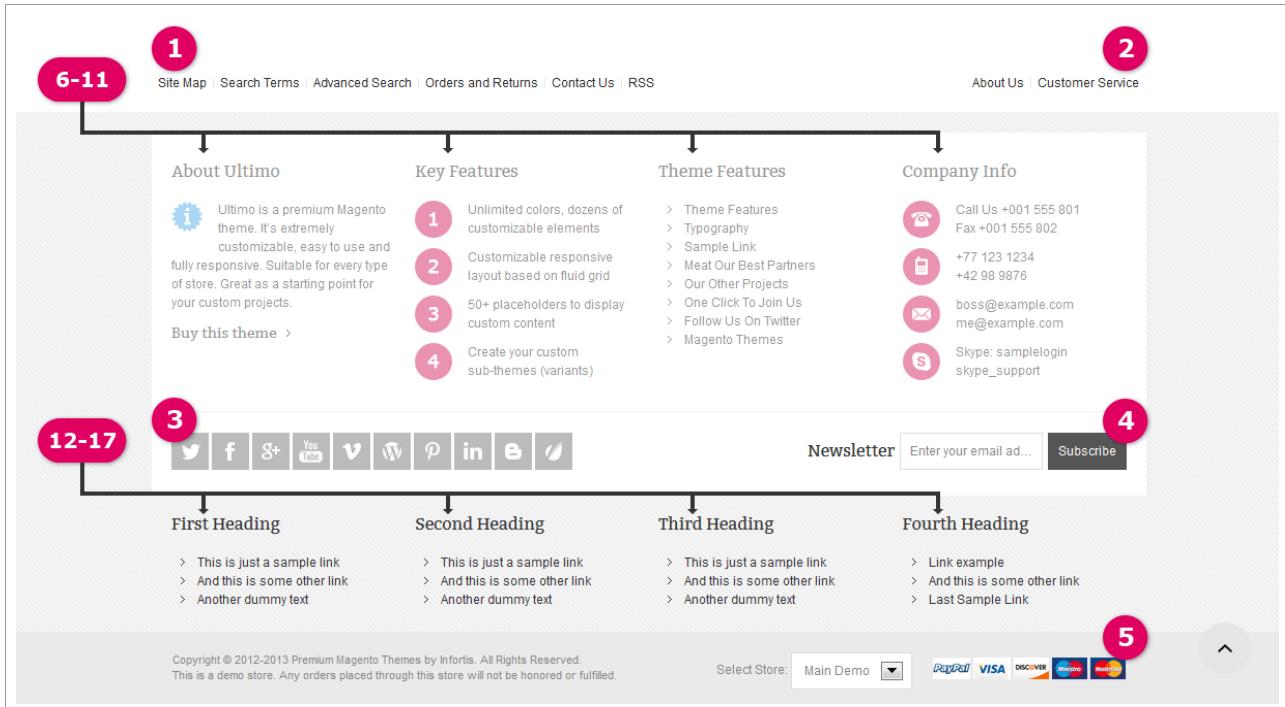
Example 2:

A few simple links aligned to the left side:

```
<div class="item item-left">
    <ul class="links">
        <li>
            <a href="#">Mobile</a>
        </li>
        <li>
            <a href="#">Links</a>
        </li>
    </ul>
</div>
```

6.2.3 Footer

Static blocks with the following identifiers will be displayed in the footer section of the page:



1. **block_footer_links** – this block can substitute Magento's default Footer Links (set of links: *Site Map*, *Search Terms*, *Contact Us* etc.). Default Footer Links can be disabled in the admin panel, refer to **7.9 Default Magento blocks** for more details.
2. **block_footer_links2** – additional block for footer links, floating right.
3. **block_footer_primary_bottom_left** – icons with links to social services.
4. **block_footer_primary_bottom_right** – this block can substitute Magento's default newsletter subscription form. Newsletter form can be disabled in the admin panel, refer to **7.9 Default Magento blocks** for more details.
5. **block_footer_payment** – information about available payment methods, credit cards, SSL certificates etc.
6. **block_footer_column1** – a column of links in the main area of the footer section, suitable for any custom content: social services links, shipping information,

promotions, company address etc. You can use up to six similar static blocks.

7. **block_footer_column2** – see above
8. **block_footer_column3** – see above
9. **block_footer_column4** – see above
10. **block_footer_column5** – see above
11. **block_footer_column6** – see above
12. **block_footer_row2_column1** – additional column of links in the secondary area of the footer section, suitable for any custom content: social services links, shipping information, promotions, company address etc. You can use up to six similar static blocks.
13. **block_footer_row2_column2** – see above
14. **block_footer_row2_column3** – see above
15. **block_footer_row2_column4** – see above
16. **block_footer_row2_column5** – see above
17. **block_footer_row2_column6** – see above

Note that the columns inside the footer don't need to be equal. You can arrange your content in many different ways, you have full control over those blocks.

Disable all static blocks with ID `block_footer_column[x]` (where `[x]` is a number from 1 to 6) and leave only one block enabled, e.g. this one: `block_footer_column1`.

Thanks to that, your block `block_footer_column1` will be stretched inside the footer and it will take the entire available space.

Now, inside `block_footer_column1` organize your content using built-in grid classes, for example:

```
<div class="grid-container">
    <div class="grid12-3">Column1...</div>
    <div class="grid12-2">Column2...</div>
    <div class="grid12-2">Column3...</div>
    <div class="grid12-2">Column4...</div>
```

```
<div class="grid12-3">Column5...</div>  
</div>
```

More info about grid system can be found in [6.4 Grid system](#).

6.2.4 Category view

Static blocks with the following identifiers will be displayed in the category view:

1. **block_category_above_empty_collection** – block above the products listing, displayed if there are no products in the category.

To display there a block with custom content, create a static block with identifier `block_category_above_empty_collection`.

2. **block_category_below_collection** – block below the products listing in the category view.

To display there a block with custom content, create a static block with identifier 3. `block_category_below_collection`.

Note:

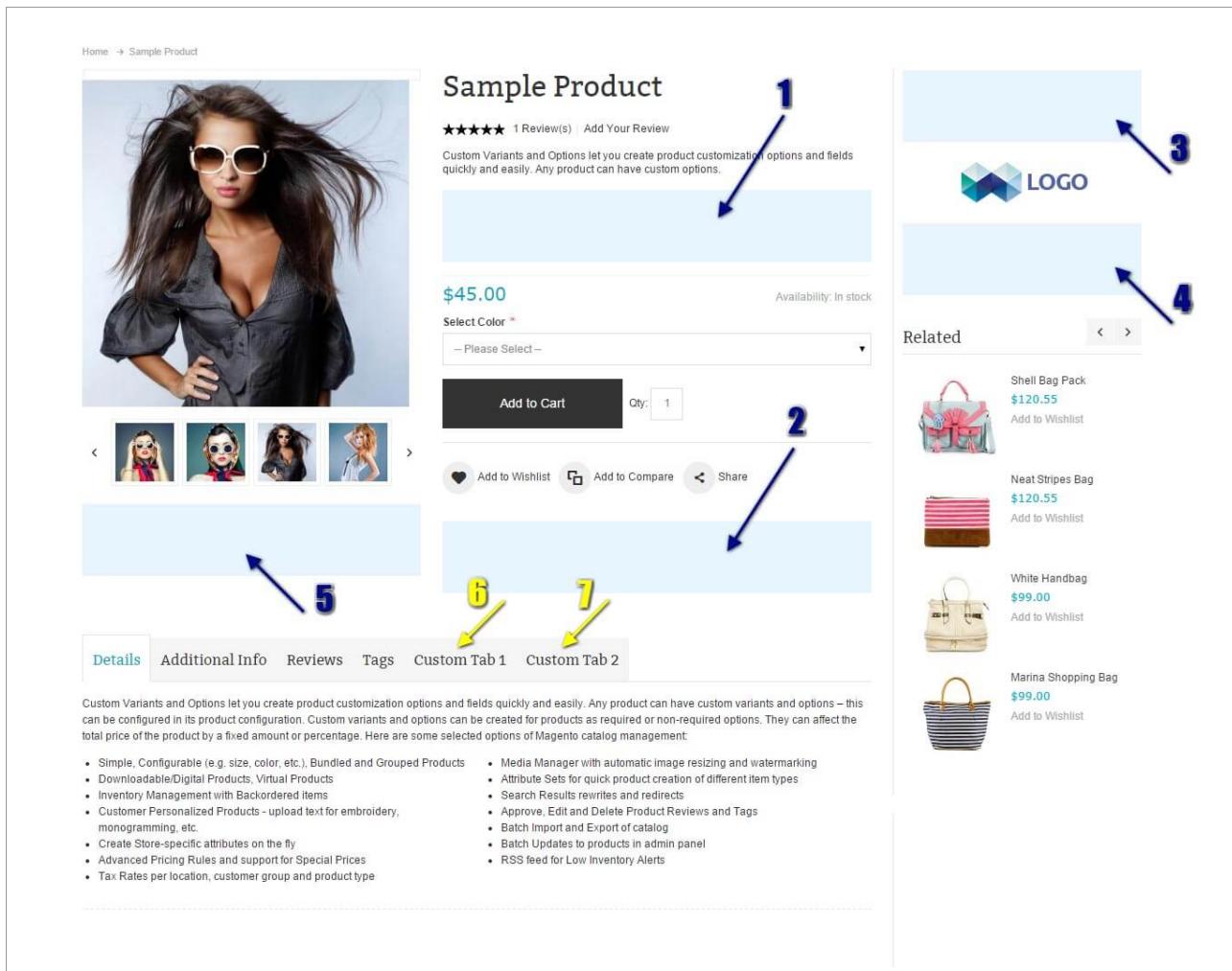
in the category view you can also add custom static blocks to the containers which are available on all pages of the store. For more information refer to [6.2.1 Page](#).

Note:

in the sidebar of the category view you can also display a block with the list of categories. For more information refer to [13.1.3 Sidebar Menu](#).

6.2.5 Product page

The screenshot below presents positions of block containers which are available on product page:



You can assign any number of custom **static blocks** to each container. If block is assigned to a container, you can choose to display it on all pages, but you can also choose to display it only on selected pages. For details about assigning static blocks to containers refer to chapter [4.8.1 Add blocks to selected pages](#).

Containers are suitable for any kind of additional information: discounts and promotions, available shipping methods, available payment methods, promotional image banners, social services bookmarks (such as AddThis bookmarks) etc.

List of containers which are available on product page:

1. **Product View, Primary Column, Container 1** – container displayed below product's short description inside the primary column.
2. **Product View, Primary Column, Container 2** – container at the bottom of the primary column.

Note:

for backward compatibility, the static block with ID **block_product_primary_bottom** (if it exists in your Magento) is displayed automatically inside this container.

3. **Product View, Secondary Column, Container 1** – container at the top of the secondary column.
4. **Product View, Secondary Column, Container 2** – container at the bottom of the secondary column.

Note:

for backward compatibility, the static block with ID **block_product_secondary_bottom** (if it exists in your Magento) is displayed automatically inside this container.

5. **Product View, Image Column, Below Images** – container displayed just below the image gallery.

Note:

for backward compatibility, the static block with ID **block_product_img_bottom** (if it exists in your Magento) is displayed automatically inside this container.

Additionally you can also display two static blocks as tabs on product page. Content of these tabs will be displayed on the product page of every product in the store.

6. **block_product_tab1** – to display custom tab on product page, create a static block with identifier **block_product_tab1**. The content of that block will be displayed inside the panel of the tab and the title of that block will be displayed as the name of

the tab.

7. **block_product_tab2** – to display second custom tab on product page, create a static block with identifier `block_product_tab2`. The content of that block will be displayed inside the panel of the tab and the title of that block will be displayed as the name of the tab.

There are also two other tabs on product page for **product-specific** info:

- **Details** – tab which displays product description. You can add there any custom content, it can be a simple text as well as a complex HTML with images, lists etc. You can also embed videos from services such as YouTube, Vimeo etc.
- **Additional Info** – tab which displays product-specific attributes. You can enable/disable each attribute's visibility on product page with **Visible on Product View Page on Front-end** field in *Catalog > Attributes > Manage Attributes > [select attribute]*.

6.2.6 Shopping cart

Static blocks with the following identifiers will be displayed on the shopping cart page:

1. **block_cart_below_table** – block below the main table in the shopping cart.
2. **block_cart_below_totals** – block below the shopping cart totals.

6.2.7 Cart drop-down

Static blocks with the following identifiers will be displayed in the cart drop-down box:

1. **block_mini_cart_above_products** – block inside cart drop-down, suitable for a short info about promotions, sales etc.

6.3 Pages and elements

6.3.1 Product page

6.3.1.A Social bookmarks (AddThis)

To display social bookmarks this theme uses the AddThis widget. The code of the widget can be found on this site: <http://www.addthis.com/get>

You can generate many different versions of the widget code to create customized bookmarks. Then, add the code to your static block or page and **flush Magento cache**.

AddThis bookmarks can be, for example, placed in the static block with ID `block_product_secondary_bottom`. This block is positioned at the bottom of the secondary column on product page.

Important:

always turn off WYSIWYG editor when you want to edit HTML or Magento's CMS tags in the page/block content. Otherwise WYSIWYG editor will break the content. Refer to **4.8.4 WYSIWYG editor** for more details.

6.3.2 Drop-down cart

This theme is using Magento's default sidebar cart to display drop-down cart at the top of the page. In order to show drop-down cart, option **Display Shopping Cart Sidebar** in *System > Configuration > Checkout > Shopping Cart Sidebar* has to be enabled.

6.4 Grid system

In version 1.17.0 of Ultimo theme we added the grid system from Bootstrap (Bootstrap version 3.3.7). This new [Bootstrap grid system](#) will replace the old deprecated grid system which we have used in Ultimo theme since the very beginning.

Note:

we didn't add a complete Bootstrap library, but only our custom build of Bootstrap. Only these components were compiled: "Grid system" (`grid.less`) and "Responsive utilities" (`responsive-utilities.less`).

You can still use the old grid (the code will be still available), but **we recommend to use the Bootstrap grid** – especially when you create custom content (e.g. on CMS pages) or when you override (or create new) template files in your custom sub-themes. The new grid system is more powerful so we think you will enjoy the change.

6.4.1 Bootstrap grid overview

All the information about **Bootstrap grid system** can be found on the official website:
<http://getbootstrap.com/css/#grid>

In this chapter you can find only a quick introduction.

Important:

Bootstrap documentation states that rows must be placed within an element marked with the `container` class. You can ignore this information, **don't use the `container` class in your custom content**. Ultimo theme already uses the `container` class to build the base structure of the page so this class can't be used anywhere else, especially not in the custom content of CMS pages.

The only exception is when you want to create a full width page. In that case you may need to wrap the content in an additional div with the `container` class. See for more details: [6.1.2.C Full width page](#)

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Bootstrap grid system is based on **twelve columns**. Grid

columns are created by specifying the number of twelve available columns you wish to span. This means that for every block you can specify how many columns it should span (from 1 to 12). The number at the end of the class name indicates the number of spanned columns. For example, class `col-md-3` will create a block which is 3-columns wide. Which means: the width of the block equals 3/12 of the entire available space.

Here's how the Bootstrap grid system works:

- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like `row` and `col-xs-4` are available for quickly making grid layouts.
- Columns create gutters (gaps between column content) via padding attribute. That padding is offset in rows for the first and last column via negative margin on elements with the `row` class.
- The negative margin on rows makes the rows outdented. It's so that content within grid columns is lined up with non-grid content.
- Grid columns are created by specifying the number of **twelve** available columns you wish to span. For example, three equal columns would use three `col-xs-4`.
- If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.
- Grid classes apply to devices with screen widths greater than or equal to the breakpoint sizes, and override grid classes targeted at smaller devices. Therefore, e.g. applying any `col-md-*` class to an element will not only affect its styling on medium devices but also on large devices if a `col-lg-*` class is not present.

See how aspects of the grid system work across multiple devices with a handy table.

| | Extra small devices Phones (<768px) | Small devices Tablets ($\geq 768px$) | Medium devices Desktops ($\geq 992px$) | Large devices Desktops ($\geq 1200px$) |
|----------------------|---|--|--|--|
| Grid behavior | Horizontal at all times | Collapsed to start, horizontal above breakpoints | | |
| Class prefix | <code>col-xs-</code> | <code>col-sm-</code> | <code>col-md-</code> | <code>col-lg-</code> |
| Column width | Auto | ~62px | ~81px | ~97px |
| Gutter width | 30px (15px on each side of a column) | | | |

Look to the examples for applying these principles to your code.

6.4.2 Grid examples

Using a single set of `col-md-*` grid classes, you can create a basic grid system that starts out stacked on mobile devices and tablet devices (the extra small to small range) before becoming horizontal on desktop (medium) devices. Place grid columns in any `row`.

Two equal columns

Get two equal-width columns starting at desktops and scaling to large desktops. On mobile devices, tablets and below, the columns will automatically stack.

```
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

Three equal columns

Get three equal-width columns starting at desktops and scaling to large desktops. On mobile devices, tablets and below, the columns will automatically stack.

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
```

Three unequal columns

Get three columns starting at desktops and scaling to large desktops of various widths. Remember, grid columns should add up to twelve for a single horizontal block. More than that, and columns start stacking no matter the viewport.

```
<div class="row">
  <div class="col-md-3">.col-md-3</div>
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-3">.col-md-3</div>
</div>
```

Two unequal columns

Get two columns starting at desktops and scaling to large desktops.

```
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
```

6.4.3 Additional utility classes for grid system

Row without outdent

Rows in Bootstrap's grid system are outdented (with negative margin at both sides of a row). Thanks to that the content within grid columns is lined up with non-grid content. But sometimes (e.g. on CMS pages) you may need a row with no outdent so that the edges (not the content inside) of the grid columns are lined up with non-grid content. You can force this behavior with the additional class `row-no-outdent` which will remove the outdent from the elements with the `row` class:

```
<div class="row row-no-outdent">
  <div class="col-md-6">Column example</div>
  <div class="col-md-6">Column example</div>
</div>
```

Row with bottom gutter

To add a bottom gutter to all grid columns inside a row, add the `row-bottom-gutter` class to the row element. This is an easy way to add equal spaces between rows. The bottom gutter will have the same size as the gutter between columns. Here's an example:

```
<div class="row row-bottom-gutter">
  <div class="col-md-4">Column example</div>
  <div class="col-md-4">Column example</div>
  <div class="col-md-4">Column example</div>
</div>
<div class="row">
  <div class="col-md-4">Column example</div>
  <div class="col-md-4">Column example</div>
```

```
<div class="col-md-4">Column example</div>  
</div>
```

If the bottom gutter should be narrower, use the `row-bottom-gutter-half` class – the bottom gutter will be half the size of the gutter between columns:

```
<div class="row row-bottom-gutter-half">  
    <div class="col-md-4">Column example</div>  
    <div class="col-md-4">Column example</div>  
    <div class="col-md-4">Column example</div>  
</div>  
<div class="row">  
    <div class="col-md-4">Column example</div>  
    <div class="col-md-4">Column example</div>  
    <div class="col-md-4">Column example</div>  
</div>
```

Column without gutter

Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on rows.

But sometimes you may need to remove the gutter from selected grid columns. You can remove the gutter by adding the `no-gutter` class to the columns:

```
<div class="row">  
    <div class="col-md-6 no-gutter">Column example</div>  
    <div class="col-md-6 no-gutter">Column example</div>  
</div>
```

To remove only the gutter at the left side, use the `no-left-gutter` class. To remove only the gutter at the right side, use the `no-right-gutter` class.

6.4.4 Old 12-column grid system (deprecated)

This grid system is used only for theme interface elements. If you want to use a grid in

your custom content (e.g. on CMS pages), we recommend to use the **Bootstrap grid system** which we added to the theme in version 1.17.0 – it is described in chapter:

6.4.1 Bootstrap grid overview.

Below you can find description of the old deprecated grid system.

Theme interface is based on 12-column grid system. This means that for every element you can specify width in grid units (from 1 unit to 12 units) using grid classes. These are available classes:

```
grid12-1  
grid12-2  
grid12-3  
grid12-4  
grid12-5  
grid12-6  
grid12-7  
grid12-8  
grid12-9  
grid12-10  
grid12-11  
grid12-12
```

The number at the end of the class name indicates number of grid units. For example, class `grid12-3` will create a block which is 3-units wide.

As this is a 12-column grid, each grid class spans a number of those 12 columns, and should always add up to 12 for each row.

Simple grid examples

2 columns

If you want to display content on your custom page in two equal columns, create two blocks (`div`) and add appropriate class with number of grid units. In this case use `grid12-6` class to create two 6-units wide blocks. And $6 + 6 = 12$, so you will create a row of two equal columns.

Then wrap the blocks inside another block with class `grid-container` – this class is important to clear floats of the columns (all columns float to the left side) so that the content of the grid doesn't overlap with other content below and above the grid. Here's an example:

```
<div class="grid-container show-grid">
```

```
<div class="grid12-6">Sample text...</div>
<div class="grid12-6">Sample text...</div>
</div>
```

When developing the grid, you can also temporarily add class `show-grid` to visualize the grid (like in the above example).

3 columns

```
<div class="grid-container">
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
</div>
```

4 columns

```
<div class="grid-container">
  <div class="grid12-3">Sample text...</div>
  <div class="grid12-3">Sample text...</div>
  <div class="grid12-3">Sample text...</div>
  <div class="grid12-3">Sample text...</div>
</div>
```

3 columns (not equal)

Of course columns doesn't have to be equal, you can use grid units in different proportions. But the sum of all units in one row should be equal to 12, otherwise some columns may drop to another row. Here is an example of three columns which are not equal:

```
<div class="grid-container">
  <div class="grid12-7">Sample text...</div>
  <div class="grid12-3">Sample text...</div>
  <div class="grid12-2">Sample text...</div>
</div>
```

Multiple rows

To create multiple rows of columns simply add more units to the container:

```
<div class="grid-container">
  <div class="grid12-6">Sample text...</div>
  <div class="grid12-6">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
</div>
```

Space between rows

To add even vertical spaces between rows, replace `grid-container` with `grid-container-spaced`:

```
<div class="grid-container-spaced">
  <div class="grid12-6">Sample text...</div>
  <div class="grid12-6">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
  <div class="grid12-4">Sample text...</div>
</div>
```

Grid on mobile devices

If the web browser's viewport width is **below 768 px**, all grid columns are stretched (they are as wide as the container) and stack vertically. If you want to avoid this and want your columns to stay narrow on narrow screens, you can add special class `mobile-grid` to grid classes. It can be added to the following grid classes: `grid12-1`, `grid12-2`, `grid12-3`.

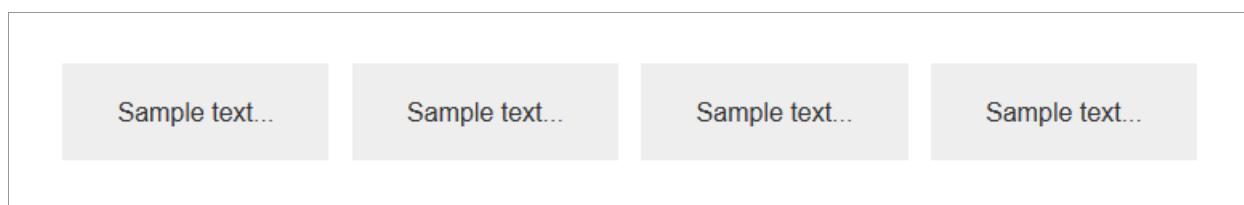
On narrow screens columns could be too narrow to display content properly, so thanks to the `mobile-grid` class, each column will be 2 times wider than the actual number of units in the class name, i.e. `grid12-3` will behave like `grid12-6`.

When the viewport width goes **below 480 px**, all grid columns with the `mobile-grid` class will also be stretched and stacked vertically, just like all the other columns.

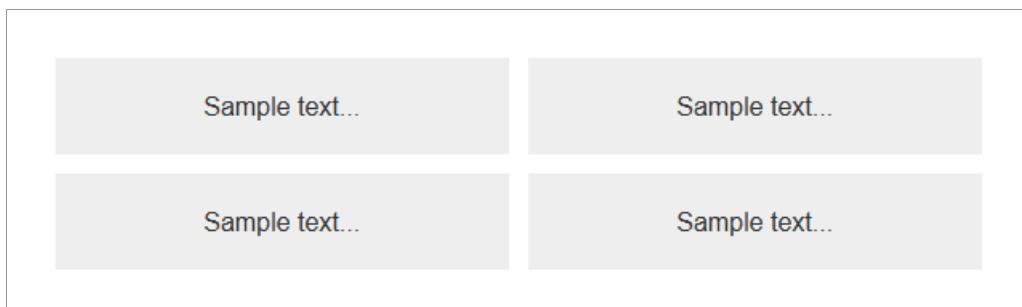
The following example will create 4 equal columns:

```
<div class="grid-container-spaced">
  <div class="grid12-3 mobile-grid">Sample text...</div>
  <div class="grid12-3 mobile-grid">Sample text...</div>
  <div class="grid12-3 mobile-grid">Sample text...</div>
  <div class="grid12-3 mobile-grid">Sample text...</div>
</div>
```

which will look like this:



When the viewport is below 768 px, the 4 columns will be automatically transformed to 2 rows of 2 columns:



6.5 CMS components

In this chapter you can find description of selected components which can be used to build custom content for CMS pages and static blocks.

6.5.1 CSS utility classes

This chapter presents available utility classes (also called helper classes). Utility classes are simple CSS classes which can be used to modify HTML content of any page or static block. You can use utility classes as an alternative for inline CSS styles.

6.5.1.A General utility classes

Here you can find list of single responsibility classes – each class is responsible for doing one job.

For example, to add a bottom margin below selected text, add the `margin-bottom` utility class to the paragraph element:

```
<p class="margin-bottom">This is a sample text.</p>
Second line of text.
```

If the text is not placed inside any element, wrap the text inside a `<div></div>` element and add the class to that `<div>`:

```
<div class="margin-bottom">
    This is a sample text.
</div>
Second line of text.
```

You can use many classes on a single element. For example, to make the text aligned to the center, add the `text-center` class:

```
<div class="margin-bottom text-center">
```

```
This is a sample text.  
</div>  
Second line of text.
```

All available classes are presented below.

Floating

- `pull-left` – float element to the left. This class is an equivalent of the CSS property `float: left`. It specifies that an element should be placed along the left side of its container, where text and inline elements will wrap around it. This way, the element is taken from the normal flow of the web page, though still remaining a part of the flow.
- `pull-right` – float element to the right. This class is an equivalent of the CSS property `float: right`.
- `clearfix` – clear floats. Add this class to the container (the parent element) of the floating elements.

Margin

- `margin-top` – add margin to the top.
- `margin-bottom` – add margin to the bottom.
- `no-margin` – remove all margin.

Padding

- `padding-top` – add padding to the top.
- `padding-bottom` – add padding to the bottom.
- `no-padding` – remove all padding.

Text

- `text-left` – align text to the left.
- `text-right` – align text to the right.
- `text-center` – align text to the center.

Design

- `no-bg` – remove background image from the element.

6.5.1.B Hide/show content depending on screen size

In version 1.17.0 of Ultimo theme we added new utility classes for showing and hiding content depending on the width of the screen. These classes come from the Bootstrap library (Bootstrap version 3.3.7). They will replace the old deprecated classes which we have used in Ultimo theme since the very beginning.

You can still use the old classes, but we highly recommend to use the new classes when you create custom content, e.g. on CMS pages. The new classes offer much more flexibility.

All the information about new utility classes from Bootstrap can be found on the official website: <http://getbootstrap.com/css/#responsive-utilities>

In this chapter you can find only a quick introduction.

Available classes

Try to use these on a limited basis and avoid creating entirely different versions of the same site. Instead, use them to complement each device's presentation.

Use a single or combination of the available classes for toggling content across viewport breakpoints.

| | Extra small devices Phones (<768px) | Small devices Tablets (≥768px) | Medium devices Desktops (≥992px) | Large devices Desktops (≥1200px) |
|----------------------------|---|--|--|--|
| <code>.visible-xs-*</code> | Visible | Hidden | Hidden | Hidden |
| <code>.visible-sm-*</code> | Hidden | Visible | Hidden | Hidden |
| <code>.visible-md-*</code> | Hidden | Hidden | Visible | Hidden |
| <code>.visible-lg-*</code> | Hidden | Hidden | Hidden | Visible |
| <code>.hidden-xs</code> | Hidden | Visible | Visible | Visible |
| <code>.hidden-sm</code> | Visible | Hidden | Visible | Visible |
| <code>.hidden-md</code> | Visible | Visible | Hidden | Visible |
| <code>.hidden-lg</code> | Visible | Visible | Visible | Hidden |

The `.visible-**` classes for each breakpoint come in three variations, one for each CSS

`display` property value listed below.

| Group of classes | CSS display |
|--|-------------------------------------|
| <code>.visible-* - block</code> | <code>display: block;</code> |
| <code>.visible-* - inline</code> | <code>display: inline;</code> |
| <code>.visible-* - inline-block</code> | <code>display: inline-block;</code> |

So, for extra small (`xs`) screens for example, the available `.visible-* - *` classes are: `.visible-xs-block`, `.visible-xs-inline`, and `.visible-xs-inline-block`.

Example 1

To hide an element, we can add a class to that element. For example, with the `.hidden-xs` class, this element will be hidden on **extra small** devices – when viewport width is below 768 pixels:

```
<div class="hidden-xs">
    This is sample text. It will be hidden on extra small devices.
</div>
```

Example 2

This element will be hidden on **small** devices – when viewport width is equal or larger than 768 pixels but below 992 pixels:

```
<div class="hidden-sm">
    This is sample text. It will be hidden on small devices.
</div>
```

Example 3

In this example we combine two classes: `.hidden-xs` and `.hidden-sm`. This element will be hidden on **small** and **extra small** devices – when viewport width is below 992 pixels:

```
<div class="hidden-xs hidden-sm">  
    This is sample text. It will be hidden on small and extra small devices.  
</div>
```

6.5.1.C Old classes to hide elements (deprecated)

These are old **deprecated** classes. If you want to hide elements below specific screen resolution in your custom content (e.g. on CMS pages), we recommend to use the new utility classes which we added to the theme in version 1.17.0 – the classes are described in previous chapter: **6.5.1.B Hide/show content depending on screen size**.

Below you can find description of the old deprecated classes.

These classes can hide elements below specific screen resolution. Number at the end of the class name indicates the resolution:

```
hide-below-1680  
hide-below-1440  
hide-below-1360  
hide-below-1280  
hide-below-960  
hide-below-768  
hide-below-480  
hide-below-320
```

For example, add class `.hide-below-1280` to the HTML element which needs to be hidden if the screen is less than 1280 pixels wide.

```
<div class="hide-below-1280">  
    This is just a sample text  
</div>
```

6.5.2 Collapsible blocks

With collapsible block, your content can be shown and hidden on button click, you can build components similar to accordions. Blocks like this are used in the footer static blocks in the demo:

```
<div class="collapsible">
    <h6 class="block-title active">My block title</h6>
    <div class="block-content">Sample content</div>
</div>
```

If you remove `active` class, the block will be collapsed by default.

If you want the block to be collapsed **only** on mobile devices (with viewport width below 768 pixels), use `mobile-collapsible` class instead of `collapsible` class in the main wrapper:

```
<div class="mobile-collapsible">
    <h6 class="block-title">My block title</h6>
    <div class="block-content">Sample content</div>
</div>
```

6.5.3 Icons

Icons are a simple and effective way to draw users into the content of your website. They can help you structure content and separate different sections of the page. The primary goal of using icons should be to help the user find information on the page.

In this chapter you can find description and examples of all kinds of icons. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.12 Data Import** for more details about the import procedure. Identifier of the page with examples of icons is: `magento-icons`.

6.5.3.A Basic icons

You can place icons just about anywhere using simple markup. We are going to use an inline HTML element such as `` and add appropriate classes to it. These are required classes: `ic` and the icon's name prefixed with `ic-`, for example `ic-star`. Here's the full example of the code which will add star icon:

```
<span class="ic ic-star"></span>
```

If you change the font-size of the icon's container, the icon gets bigger. Same things goes for color, drop shadow, and anything else that gets inherited using CSS.

Note:

list of names of all available icons can be found in
`Resources/Icons_Demo/demo.html` inside the theme package.

6.5.3.A.1 Add CSS styles to icons

Now you can start having more fun with icons. By default all icons have the same color as text, but if you want to change the color of selected icon, you can do it with **inline styles**. Add `style` attribute to the icon element and specify the color. For example, this will make the icon red:

```
<span class="ic ic-star" style="color:red;"></span>
```

You can add inline styles to icons the same way as to any other HTML elements in a HTML document. The `style` attribute can contain any CSS property, such as font-size, text-shadow etc. List of available CSS properties can be found on <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

6.5.3.A.2 Size

If you change the font-size of the icon's container, the icon gets bigger. To increase icon size relative to the font-size of the icon's container, use the following classes:

`ic-lg` (increases the size of the icon by 33%), `ic-2x`, `ic-3x`, `ic-4x`, `ic-5x`, `ic-6x`, `ic-7x`, `ic-8x`.

```
<span class="ic ic-star ic-lg"></span>
<span class="ic ic-twitter ic-2x"></span>
<span class="ic ic-heart-o ic-3x"></span>
```

To have more control over the size of the icon, you can also change the font size of the icon element by adding inline styles:

```
<span class="ic ic-star" style="font-size:32px;"></span>
```

6.5.3.A.3 List icons

Use `ic-ul` and `ic-li` to build unordered lists with icons displayed as bullets.

```
<ul class="ic-ul">
  <li><span class="ic ic-star ic-li"></span>Some text example</li>
  <li><span class="ic ic-tag ic-li"></span>Some text example</li>
  <li><span class="ic ic-arrow-right ic-li"></span>Some text example</li>
  <li><span class="ic ic-letter ic-li"></span>Some text example</li>
  <li><span class="ic ic-bell ic-li"></span>Some text example</li>
</ul>
```

6.5.3.A.4 Pulled icons and borders

Use `ic-border` and `ic-pull-left` or `ic-pull-right` for easy pull article icons.

```
<span class="ic ic-alarm ic-3x ic-border pull-left "></span> Some sample text. Lid est laborum
et dolorum fuga. Et harum quidem rerum facilis est et expeditasi distinctio.
```

6.5.3.A.5 Spinning icons

Use the `ic-spin` class to get any icon to rotate.

```
<span class="ic ic-reload ic-spin"></span>
```

6.5.3.A.6 Rotated and flipped

To rotate and flip icons, use the `ic-rotate-*` and `ic-flip-*` classes.

```
<span class="ic ic-twitter"></span> normal<br>
```

```
<span class="ic ic-twitter ic-rotate-90"></span> ic-rotate-90<br>
<span class="ic ic-twitter ic-rotate-180"></span> ic-rotate-180<br>
<span class="ic ic-twitter ic-rotate-270"></span> ic-rotate-270<br>
<span class="ic ic-twitter ic-flip-horizontal"></span> ic-flip-horizontal<br>
<span class="ic ic-twitter ic-flip-vertical"></span> ic-flip-vertical
```

6.5.3.A.7 Icons inside buttons

Font icons work great inside buttons and other elements.

```
<a class="button" href="#"><span class="ic ic-user"></span>&nbsp; My Account</a>
```

6.5.3.B Iconbox (icon with background color)

To display an icon inside a box with background color (we call it an **iconbox**), add the **ib** class to the icon element:

```
<span class="ic ic-star ib"></span>
```

Background color will be automatically added to the icon element. Make sure to leave the **** tag empty – otherwise the contents of the tag will be displayed together with the icon inside the iconbox and any additional space can dislocate the icon.

Note:

The default colors of the iconbox can be configured in the admin panel:
System > Configuration > Theme Design > Colors > Iconbox

6.5.3.B.1 Change color on mouse hover

To change the color on mouse hover over the iconbox, add the `ib-hover` class to the iconbox.

```
<span class="ic ic-star ib ib-hover"></span>
```

6.5.3.B.2 Size

To increase iconbox size, use the following classes:

`ib-size-l`, `ib-size-xl`, `ib-size-xxl`, `ib-size-xxxl`. For example:

```
<span class="ic ic-star ib ib-size-xxl"></span>
```

The size of the icon is independent of the iconbox and can be increased with classes which were described earlier in this chapter. For example, add class `ic-2x` to make the icon a little bit bigger:

```
<span class="ic ic-star ic-2x ib ib-size-xxl"></span>
```

6.5.3.B.3 Shape

To change the shape of the iconbox, use one of the following classes:

`ib-circle`, `ib-rounded`, `ib-square`. For example:

```
<span class="ic ic-star ib ib-square"></span>
```

By default the iconbox is circular. If you prefer square icons, you can wrap multiple icons with an element (it can be `<div>`) with class `ib-wrapper-square`:

```
<div class="ib-wrapper-square">
    <span class="ic ic-star ib"></span>
    <span class="ic ic-heart ib"></span>
    <span class="ic ic-letter ib"></span>
</div>
```

6.5.3.B.4 Iconbox effects

To add eye-catching hover effects to the iconbox, use one of the following combinations of classes:

- `ib-ef-1 ib-ef-1a`
- `ib-ef-1 ib-ef-1b`
- `ib-ef-2 ib-ef-2a`
- `ib-ef-2 ib-ef-2b`
- `ib-ef-3 ib-ef-3a`
- `ib-ef-3 ib-ef-3b`

Note that in each case the combination consists of two classes. For example:

```
<span class="ic ic-star ib ib-ef-1 ib-ef-1a"></span>
```

6.5.3.B.5 Image file iconbox

You can also use image files to create iconboxes with custom icons. With the following markup: `` tag with classes `ic ic-img ib` and `` tag inside the `` tag, you can quickly create custom icon:

```
<span class="ic ic-img ib">
    
</span>
```

To get the URL of the image, you need to use Magento's CMS tag `media`. Magento will replace the tag with the URL of the image, for example:

`{{media url="wysiwyg/path/to/image.png"}}`. The path is relative to Magento's `media` directory, so you need to store your images inside that directory.

Recommended size of an image: 30 x 30 pixels. An image can be larger but it will be scaled down to fit the size of the iconbox. So it is better to use images of correct size to keep good quality.

6.5.3.C Social icons

To add icons of social networking services you need to use exactly the same markup as for other icons described earlier in this chapter. Name of the service needs to be prefixed with `ic-`, for example `ic-twitter`. Here's the full example of the code which will add a Twitter icon:

```
<span class="ic ic-twitter"></span>
```

Note:

list of names of all available icons can be found in
Resources/Icons_Demo/demo.html inside the theme package.

Icons can be wrapped inside links. This is very useful when you want to create links to your profiles on social networking services. Here's an example:

```
<a href="https://twitter.com/infortis">
    <span class="ic ic-twitter"></span>
</a>
```

You can also add iconbox classes (`ib` and `ib-hover`, refer to **6.5.3.B Iconbox (icon with background color)** for more details) to create an icon with background color, for example:

```
<a href="https://twitter.com/infortis">
    <span class="ib ib-hover ic ic-twitter"></span>
</a>
```

If you additionally wrap the icon inside a container with the `social-links` class, you will be able to configure colors through the admin panel. The default colors for social icons can be configured under:

System > Configuration > Theme Design > Colors > Iconbox (Social Icons)

In the example below, replace `#` with URLs of your profiles on social networking services:

```
<div class="social-links">
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-twitter"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-facebook"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-googleplus"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-youtube"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-vk"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-renren"></span>
    </a>
    <a href="#">
        <span class="ib ib-hover ic ic-lg ic-instagram"></span>
    </a>
</div>
```

6.5.3.D Letters, numbers and other characters as icons

By adding class `ic-char` to the icon, you can display any characters (such as letters, numbers and other special characters) as icons. The code is almost the same as for standard font icons, but inside the `` tag you need to add a single character. For example:

```
<span class="ic ic-char ic-2x ib">1</span>
<span class="ic ic-char ic-2x ib">2</span>
<span class="ic ic-char ic-2x ib">A</span>
<span class="ic ic-char ic-2x">@</span>
<span class="ic ic-char ic-2x">$</span>
<span class="ic ic-char ic-2x">%</span>
```

6.5.3.E Font Awesome icons

Font Awesome is a font and icon toolkit based on CSS. It offers a collection of more than 600 vector icons which can be easily customized (the same as other font icons available in the theme).

Font Awesome can be enabled in admin panel under *System > Configuration > Theme Settings > Integration > Font Awesome*.

All the information about Font Awesome can be found on the official website:

<http://fontawesome.io/examples/>

In this chapter you can find only a quick introduction.

Basic Font Awesome icons

To create an icon, use the `fa` class and the icon's name with an inline HTML element ``. Here's an example of the code which will create a flag icon:

```
<span class="fa fa-flag fa-3x"></span>
```

Use Font Awesome icons with other icon classes

You can use Font Awesome icons together with other icon classes described in this document. Here's an example of an **iconbox** element (the **ib** class) with Font Awesome icon inside a block:

```
<div class="feature feature-icon-hover indent indent-size-1">
  <span class="ic ic-2x ib ib-size-1 left fa fa-flag"></span>
  <h4>Heading Example</h4>
  <p>This is a short paragraph of sample text inside a block.</p>
</div>
```

6.5.4 Icon + text

Icons can help you structure content and separate different sections of the page. The primary goal of using icons should be to help the user find information on the page and with icons you can quickly sum up what your text is about. For example, when you build lists, instead of using standard bullets, you can use icons to draw attention to paragraphs and other blocks of content.

In this chapter you can find information about creating blocks of text with icons. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.12 Data Import** for more details about the import procedure. Identifier of the page with examples is: [magento-icons](#).

6.5.4.A Simple block with icon

To create a simple block of text with an icon, wrap your text inside a `<div>` element with the `feature` class. Here's a minimal example:

```
<div class="feature">
    <span class="left ic ic-star"></span>
    <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
    of blocks.</p>
</div>
```

If you add `left` or `right` class to the icon element, the icon will be taken from the normal flow and placed along the left or right side of its container, and text will wrap around it.

Using this markup you can quickly build all kinds of blocks (with text paragraphs, headings, links etc.) enhanced with icons. For more information about creating icons, refer to chapter **6.5.3 Icons**.

Here's another, more complex example with additional headings and a bigger icon inside an iconbox:

```
<div class="feature">
```

```
<span class="left ic ic-star ic-2x ib ib-size-xl"></span>
<h6 class="above-heading">Above heading</h6>
<h3>Block Heading Example</h3>
<h6 class="below-heading">Text below heading</h6>
<p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
<p>Example of another text paragraph inside a block. Icons are an effective way to draw
users into the content of your store.</p>
<a href="#">Read more...</a>
</div>
```

6.5.4.B Change iconbox color on mouse hover

To change the background color of the **iconbox** on mouse hover over the entire block, add the `feature-icon-hover` class to the block element:

```
<div class="feature feature-icon-hover">
<span class="left ic ic-star ib"></span>
<h3>Block Heading Example</h3>
<p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

The default colors of the iconbox can be configured in the admin panel.
Refer to **6.5.3.B Iconbox (icon with background color)** for details.

6.5.4.C Indented block

To display a block with indentation on the left side, add the `indent` class to the block element:

```
<div class="feature feature-icon-hover indent">
<span class="left ic ic-star ib"></span>
<h3>Block Heading Example</h3>
<p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
of blocks.</p>
</div>
```

If you increase the iconbox size (by adding a class such as `ib-size-xxl`), you will also need to add corresponding class (in this case: `indent-size-xxl`) to the block element. It will adjust the size of the indentation, for example:

```
<div class="feature feature-icon-hover indent indent-size-xxl">
  <span class="left ic ic-star ib ib-size-xxl"></span>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
  of blocks.</p>
</div>
```

To change the size of the indentation, use the following classes:

`indent-size-l`, `indent-size-xl`, `indent-size-xxl`, `indent-size-xxxl`.

6.5.4.D Centered block

To align elements of the block to the center, use class `centered`. Like this:

```
<div class="feature centered">
  <span class="ic ic-star ib ib-size-xl"></span>
  <h3>Block Heading Example</h3>
  <p>This is a paragraph of sample text. Using this markup you can quickly build all kinds
  of blocks.</p>
</div>
```

6.5.5 Banners

Image banners can be useful in many different areas of your store. You can place banners just about anywhere using simple markup. With additional utility classes you can easily add captions and multiple hover effects to your banners.

In this chapter you can find description and examples of all kinds of banners. You can also **import a sample CMS page which contains dozens of ready-to-use code examples**. Refer to chapter **7.12 Data Import** for more details about the import procedure. Identifier of the page with sample banners is: `magento-banners`.

6.5.5.A Image banner

The image and the entire content of the banner is wrapped inside an element with the `ban` class. The image element needs to be marked with the `image` class. Here's a minimal code example to create a simple banner:

```
<div class="ban">
    
</div>
```

6.5.5.A.1 How to get the image

To get the URL of the image, you need to use the **Media URL shortcode**. Magento will automatically replace it with the image URL during page rendering.

The image path is relative to Magento's `media` directory, so you need to store your images inside that directory.

For example, if this is the path of your image: `media/wysiwyg/path/to/image.png`, you can get the URL of that image with the following shortcode:

```
{{media url="wysiwyg/path/to/image.png"}}
```

Here's how you can use that shortcode in the `src=""` attribute of the `` element to display your image on a page:

```

```

For more details about the **Media URL shortcode**, refer to chapter: [4.8.3.B Media URL](#).

6.5.5.B Banner with hyperlink

To create a banner with a hyperlink to other pages (or to locations within the same page), wrap the banner inside the anchor element `<a>`. Inside the `href=""` attribute put a URL that the hyperlink points to. You can wrap the entire banner, or only the caption.

To get the URL of the page, you need to use the **Store URL shortcode**. For example:

```
<a href="{{store url="about-us"}}>
    <div class="ban">
        
    </div>
</a>
```

Replace `about-us` with the **URL Key** of any page in your store. Path is relative to the store URL. For example, if you want to create a link to the category page which have the following URL in your store: www.example.com/cell-phones, simply replace `about-us` with `cell-phones`:

```
<a href="{{store url="cell-phones"}}>
    <div class="ban">
        
    </div>
```

```
</a>
```

For more details about the **Store URL shortcode**, refer to chapter: [4.8.3.A Store URL](#).

6.5.5.C Banner with image hover effect

To add a hover effect to the banner, add an effect class to the banner element (to the element with the `ban` class). The following classes are available: `ban-effect-1`, `ban-effect-2`, `ban-effect-3`, `ban-effect-fade-out`, `ban-effect-grayscale-in`, `ban-effect-grayscale-out`, `ban-effect-blur-in`, `ban-effect-blur-out`.

```
<div class="ban ban-effect-1">
    
</div>
```

6.5.5.D Banner with color overlay

To add a color overlay to the banner, add a caption but without any text. Captions are described with more details in chapter [6.5.5.E Banner with caption](#), but in this section we will only use an empty caption element.

To add an empty caption to the banner, simply add an element with the `cap` class inside the banner: `<div class="cap"></div>`. The empty caption element will be displayed as a color overlay. Here's an example:

```
<div class="ban">
    
    <div class="cap"></div>
</div>
```

Note:

The default color of the overlay can be configured in the admin panel:
System > Configuration > Theme Design > Colors

Change the color of the overlay

To change the color of the overlay, you can also add an inline CSS to the caption element to override the default value of its `background-color` property. You can add semi-transparent color using the `rgba()` functional notation allowing specification of the opacity of a color. For example:

```
<div class="ban">
    
    <div class="cap" style="background-color: rgba(135, 195, 30, 0.45);"></div>
</div>
```

Color overlay with fade in/out effect

By default the color overlay (the caption element) covers the banner all the time. But you can change it with two useful classes. The `ban-caption-fade-out` class will make the overlay fade out on mouse hover over the banner. And the `ban-caption-fade-in` class will make the overlay fade in only on mouse hover over the banner. Add these classes to the banner element, **not** to the caption element.

```
<div class="ban ban-caption-fade-out">
    
    <div class="cap" style="background-color: rgba(245, 88, 86, 0.75);"></div>
</div>

<div class="ban ban-caption-fade-in">
    
    <div class="cap" style="background-color: rgba(0, 0, 0, 0.85);"></div>
</div>
```

6.5.5.E Banner with caption

To create a caption for the banner, add an element with the `cap` class inside the banner. Inside the caption you can add any text elements, such as headings or paragraphs of text. The caption (with a color overlay) will be displayed over the image.

```
<div class="ban">
    
    <div class="cap">
        <h3>Caption Example</h3>
        <p>This is a sample text</p>
    </div>
</div>
```

Note:

The default color of the overlay can be configured in the admin panel:
`System > Configuration > Theme Design > Colors`

You can extend this simple example with additional features:

- by adding more CSS classes (described in this chapter) to the banner and to the caption element
- by adding more HTML (e.g. icons) inside the caption

6.5.5.E.1 Inline styles

If you want to change the style of the caption, e.g. change the font size or the color, you can do it with **inline CSS styles**. Add the `style` attribute to the caption element and specify values of selected properties such as `font-size`, `font-family`, `color`, `background-color` etc.

In the example below, we override the default values of the `color` and the `background-color` properties of the caption element and the `font-size` property of `<h2>` and `<p>` elements. To make the background color semi-transparent, use the `rgba()` functional notation.

```
<div class="ban">
    
    <div class="cap" style="color: #000; background-color: rgba(255, 254, 84, 0.6);">
        <h2 style="font-size: 32px;"><span class="ic ic-lightbulb"></span> Caption Example</h2>
        <p style="font-size: 18px;">Create responsive banners in your Magento with ease.</p>
    </div>
</div>
```

6.5.5.E.2 Remove caption background color

To get rid of the background color from the caption, add the `cap-no-bg` class to the caption element. This will make the caption element transparent, only the text will be visible over the banner:

```
<div class="ban ban-effect-1">
    
    <div class="cap cap-no-bg">
        <h2 style="font-size: 32px;"><span class="ic ic-star"></span> Caption Example</h2>
        <p style="font-size: 18px;">Create responsive banners in your Magento with ease.</p>
    </div>
</div>
```

6.5.5.E.3 Caption with fade in/out effect

By default the caption element covers the banner all the time. But you can change it with two useful classes. The `ban-caption-fade-out` class will make the caption fade out on mouse hover over the banner. And the `ban-caption-fade-in` class will make the caption fade in on mouse hover over the banner.

Note:

Add these classes to the banner element, **not** to the caption element.

Here's the code example of the caption with fade out effect:

```
<a href="#">
    <div class="ban ban-caption-fade-out">
        
        <div class="cap">
            <h4>This caption will fade out on hover.</h4>
        </div>
    </div>
</a>
```

6.5.5.E.4 Center caption

To center the caption horizontally, add the `cap-center-horizontally` class to the caption element. To center the caption vertically, add the `cap-center-vertically` class to the caption element. Both classes can be used together.

Here's the code example of the caption centered horizontally and vertically:

```
<div class="ban">
    
    <div class="cap cap-center-vertically cap-center-horizontally">
        <h2>Caption Example</h2>
        <p>Create responsive banners in your Magento with ease</p>
    </div>
</div>
```

6.5.5.E.5 Center vertically, full cover

When the caption is centered vertically, if you want the caption to cover the entire banner, you need to use **two nested caption elements**. The **inner** caption element needs to have the `cap-center-vertically` class. The additional `cap-no-bg` class will remove the background color from the inner caption element:

```
<div class="ban">
    
    <div class="cap">
        <div class="cap cap-center-vertically cap-no-bg">
            <h2>Caption Example</h2>
            <p>Create responsive banners in your Magento with ease</p>
        </div>
    </div>
```

```
</div>  
</div>
```

6.5.5.E.6 Caption positioning

To set the position of the caption, use the following classes: `cap-top`, `cap-bottom`, `cap-left`, `cap-right`, `cap-top-left`, `cap-top-right`, `cap-bottom-left`, `cap-bottom-right`.

Here's a code example of a caption placed at the bottom of the banner:

```
<div class="ban">  
      
    <div class="cap cap-bottom">  
        <h3 class="no-margin">Caption Example</h3>  
    </div>  
</div>
```

6.5.5.E.7 Precise positioning

To position the caption with more precision, you can use the classes presented below. The number at the end of each class name indicates the value (percentage) by which the caption will be moved in the selected direction. In the **vertical direction** the percentage is calculated with respect to the height of the entire banner. In the **horizontal direction** the percentage is calculated with respect to the width of the entire banner.

For example, the `cap-push-down-10` class will push the caption down by 10%. Here's the list of all available classes:

| Push down | Push up | Push left | Push right |
|-------------------------------|-----------------------------|-------------------------------|--------------------------------|
| <code>cap-push-down-5</code> | <code>cap-push-up-5</code> | <code>cap-push-left-5</code> | <code>cap-push-right-5</code> |
| <code>cap-push-down-10</code> | <code>cap-push-up-10</code> | <code>cap-push-left-10</code> | <code>cap-push-right-10</code> |
| <code>cap-push-down-15</code> | <code>cap-push-up-15</code> | <code>cap-push-left-15</code> | <code>cap-push-right-15</code> |
| <code>cap-push-down-20</code> | <code>cap-push-up-20</code> | <code>cap-push-left-20</code> | <code>cap-push-right-20</code> |

| | | | |
|------------------|----------------|------------------|-------------------|
| | | | |
| cap-push-down-25 | cap-push-up-25 | cap-push-left-25 | cap-push-right-25 |
| cap-push-down-30 | cap-push-up-30 | cap-push-left-30 | cap-push-right-30 |

Below you can see an example of a caption which is pushed up by 20% and pushed right by 15%.

```
<div class="ban">
    
    <div class="cap cap-push-up-20 cap-push-right-15">
        <h2>Example of Banner Caption</h2>
        <p>Create custom banners with captions</p>
    </div>
</div>
```

6.5.5.E.8 Caption effects

To add eye-catching hover effects for the caption, add classes listed below to the banner element:

| Fade | Slide | Push | Hinge |
|----------------------|-------------------------|-----------------------|-------------------------|
| ban-caption-fade-out | ban-caption-slide-up | ban-caption-push-up | ban-caption-hinge-up |
| ban-caption-fade-in | ban-caption-slide-down | ban-caption-push-down | ban-caption-hinge-down |
| | ban-caption-slide-left | | ban-caption-hinge-left |
| | ban-caption-slide-right | | ban-caption-hinge-right |

Note:

Add these classes to the banner element, **not** to the caption element.

Here's a code example of a banner with caption "hinge up" effect:

```
<div class="ban ban-caption-hinge-up">
    
    <div class="cap">
        <h3>Caption Example</h3>
        <p>Create custom banners with captions</p>
    </div>
</div>
```

6.5.5.E.9 Caption with text background

To create a caption with background color behind each block of text, you need to make a few steps:

1. Mark each block of text inside the caption element with the `text` class.
2. Add these two classes to the caption element:
 - `cap-no-bg` class which removes the background color from the caption element
 - `cap-text-bg` class which adds the background color to text blocks
3. Add one of these classes to the caption element to choose the tone of the text background - dark (semi-transparent black) or light (semi-transparent white):
 - `cap-text-bg-dark-1`
 - `cap-text-bg-dark-2`
 - `cap-text-bg-dark-3`
 - `cap-text-bg-light-1`
 - `cap-text-bg-light-2`
 - `cap-text-bg-light-3`

Here's an example of a caption with dark text background:

```
<div class="ban">
    
    <div class="cap cap-no-bg cap-text-bg cap-text-bg-dark-3">
        <h2 class="text">Caption Example</h2>
        <p class="text">Create customizable banners with captions</p>
        <p class="text">Responsive Magento theme</p>
    </div>
</div>
```

6.5.5.E.10 Old classes to create captions for slideshow (deprecated)

These are old deprecated classes. If you want to create feature-rich banners with captions, we recommend to use the new system of banners which was added to the theme in version 1.18.0. Refer to chapter [6.5.5 Banners](#) and [6.5.5.E Banner with caption](#) for more details.

Below you can find description of the **old deprecated classes**.

Presented code can be used to create captions inside the homepage Slideshow (inside the main slides and inside small banners at the side of the Slideshow).

To set background color, use one of the following classes: `light1`, `light2`, `light3`, `dark1`, `dark2`, `dark3`. For example:

```
<div class="caption light1">
    <p>Lorem ipsum dolor sit amet</p>
</div>
```

To position caption, use the following classes (by default captions is positioned in the bottom left corner): `top`, `top-right`, `bottom-right`. For example:

```
<div class="caption light1 top-right">
```

```
<h2 class="heading">Big Heading</h2>
<p>Lorem ipsum dolor sit amet</p>
</div>
```

Additionally, the paragraph of text inside the caption can be aligned right with class: `right`, or centered with class: `centered`. For example:

```
<div class="caption dark1 centered">
  <h2 class="heading">Big Heading</h2>
  <p>Lorem ipsum dolor sit amet</p>
</div>
```

To stretch the caption, use class: `full-width`. To make it narrower, use: `narrow`. For example:

```
<div class="caption dark3 full-width">
  <p>Lorem ipsum dolor sit amet dictum sit altum.</p>
</div>
```

On narrow screens elements of the caption will be hidden. To keep selected elements visible, use class: `permanent`. For example:

```
<div class="caption light1">
  <h2 class="heading permanent">Heading always visible</h2>
  <p>This paragraph will be hidden on mobile</p>
</div>
```

6.5.5.F Groups of banners

To create groups of banners and various layouts (with multiple rows and columns) you can use the grid system which is described in this chapter: [6.4 Grid system](#).

For example, to create two equal-width banners displayed side by side, create two columns using the grid system. Here's the markup for columns:

```
<div class="row">
    <div class="col-md-6">
        ...Column 1...
    </div>
    <div class="col-md-6">
        ...Column 2...
    </div>
</div>
```

Now, add a banner inside each column:

```
<div class="row">
    <div class="col-md-6">
        <div class="ban">
            
        </div>
    </div>
    <div class="col-md-6">
        <div class="ban">
            
        </div>
    </div>
</div>
```

6.5.6 Animations

This chapter presents Animate.css – a cross-browser library of CSS animations. With simple CSS classes this library will let you add some life to the content of your store. All the information about the library can be also found in the official documentation:

<https://github.com/daneden/animate.css>

You can use the animations, for example, to create eye-catching icons on promotional banners, but the animations are especially useful with the **Slideshow** module – all details can be found in chapter **11.3.3 Animated content of a slide**.

6.5.6.A Basic usage

To add an animation, you need to follow these steps:

1. Add the `animated` class to the element you want to animate.
2. You may also want to include the optional `infinite` class to create an infinite loop. Without it, the animation will stop after the first cycle is finished (most animations take 1 second to complete one cycle).
3. Add class which indicates the name of the animation, for example: `pulse`. Full list of available animation can be found later in this chapter: **6.5.6.B List of animations**.

Let's test it on a simple icon. Here's the code of the icon before adding animations:

```
<span class="ic ic-heart-o ic-5x"></span>
```

And here's how it should look like with our animation classes. The `pulse` class is the name of the animation, it will add pulsing effect to the icon:

```
<span class="ic ic-heart-o ic-5x animated infinite pulse"></span>
```

Animations are triggered on page load and last only a short moment so in most cases it's good idea to use the `infinite` class to create an infinite loop.

6.5.6.B List of animations

Here's the list of class names of all available animations. All the animations can be also tested on this website:

<https://daneden.github.io/animate.css/>

- `bounce`
- `flash`
- `pulse`
- `rubberBand`
- `shake`
- `headShake`
- `swing`
- `tada`
- `wobble`
- `jello`
- `bounceIn`
- `bounceInDown`
- `bounceInLeft`
- `bounceInRight`
- `bounceInUp`
- `bounceOut`
- `bounceOutDown`
- `bounceOutLeft`
- `bounceOutRight`
- `bounceOutUp`
- `fadeIn`
- `fadeInDown`
- `fadeInDownBig`
- `fadeInLeft`
- `fadeInLeftBig`
- `fadeInRight`
- `fadeInRightBig`
- `fadeInUp`
- `fadeInUpBig`
- `fadeOut`
- `fadeOutDown`
- `fadeOutDownBig`
- `fadeOutLeft`
- `fadeOutLeftBig`
- `fadeOutRight`
- `fadeOutRightBig`
- `fadeOutUp`
- `fadeOutUpBig`
- `flipInX`
- `flipInY`
- `flipOutX`
- `flipOutY`
- `lightSpeedIn`
- `lightSpeedOut`
- `rotateIn`
- `rotateInDownLeft`
- `rotateInDownRight`
- `rotateInUpLeft`
- `rotateInUpRight`
- `rotateOut`
- `rotateOutDownLeft`
- `rotateOutDownRight`
- `rotateOutUpLeft`
- `rotateOutUpRight`
- `hinge`
- `rollIn`
- `rollOut`
- `zoomIn`
- `zoomInDown`
- `zoomInLeft`
- `zoomInRight`
- `zoomInUp`
- `zoomOut`
- `zoomOutDown`
- `zoomOutLeft`
- `zoomOutRight`
- `zoomOutUp`
- `slideInDown`
- `slideInLeft`
- `slideInRight`
- `slideInUp`
- `slideOutDown`
- `slideOutLeft`
- `slideOutRight`
- `slideOutUp`

6.5.6.C Additional options

6.5.6.C.1 Animation delay

Animations are triggered immediately on page load. If you want to delay the start of the animation, you can use one of the additional `delay-*` classes presented below. The number at the end of each class name indicates the value (in seconds) by which the animation will be delayed.

For example, class `delay-3` will delay the animation start by **3 seconds**:

```
<span class="ic ic-star ic-3x animated infinite flash delay-3"></span>
```

If we add animations to more than one element, we can use different delay classes on each element to get more interesting effects. In this example, the first icon will start animation after **0.5 seconds** and the second icon will start after **1 second**:

```
<span class="ic ic-heart-o ic-3x animated infinite pulse delay-0-5"></span>
<span class="ic ic-heart-o ic-3x animated infinite pulse delay-1"></span>
```

Here's the list of all available classes:

- `delay-0-5`
- `delay-1`
- `delay-1-5`
- `delay-2`
- `delay-2-5`
- `delay-3`
- `delay-3-5`
- `delay-4`
- `delay-4-5`
- `delay-5`
- `delay-6`
- `delay-7`
- `delay-8`
- `delay-9`
- `delay-10`

6.5.6.C.2 Animation duration

Most animations take **1 second** to complete one cycle. To make an animation last longer, you can use one of the additional `duration-*` classes presented below. These classes specify the length of time that an animation should take to complete one cycle. The number at the end of each class name indicates the time (in seconds).

For example, with the `duration-3` class the animation will take **3 seconds** to complete one cycle:

```
<span class="ic ic-star ic-3x animated infinite flash duration-3"></span>
```

While with the `duration-0-5` class the animation will take only **0.5 seconds** to complete one cycle:

```
<span class="ic ic-star ic-3x animated infinite flash duration-0-5"></span>
```

Here's the list of all available classes:

- `duration-0-5`
- `duration-1`
- `duration-2`
- `duration-3`
- `duration-4`
- `duration-5`

6.6 Theme customization

This chapter was moved to: [5. Customization](#)

7. Theme settings

This chapter refers to the following section of the theme admin panel:
System > Configuration > Ultimo > Theme Settings

Important:

beginning from this chapter, each chapter refers to a specific sections of theme admin panel. Path of the admin panel section is visible at the top of each page.

7.1 Header

This chapter refers to the following section of the theme admin panel:
System > Configuration > Theme Settings > Header

List of all settings available in that section can be found later in this chapter: **7.1.3 Header configuration options**.

7.1.1 Header structure

Width of the header sections (Left Column, Central Column, Right Column) is specified in grid units. As this is a 12-column grid, each section spans a number of those 12 columns, and should always add up to 12.

Example 1:

If you want to display three equal sections (Left Column, Central Column, Right Column), sum of these three sections has to be equal 12 grid units. For example:

Left Column = 4, Central Column = 4, Right Column = 4
4 + 4 + 4 = 12
All the three sections will have equal width.



Example 2:

If you want to get rid of one section, e.g. Central Column, leave the **Central Column** field empty and make the Right Column wider. For example:

```
Left Column = 4, Central Column = leave empty, Right Column = 8  
4 + ... + 8 = 12
```

Thanks to that the Central Column will disappear and the Right Column will be 2 times wider than the Left Column:



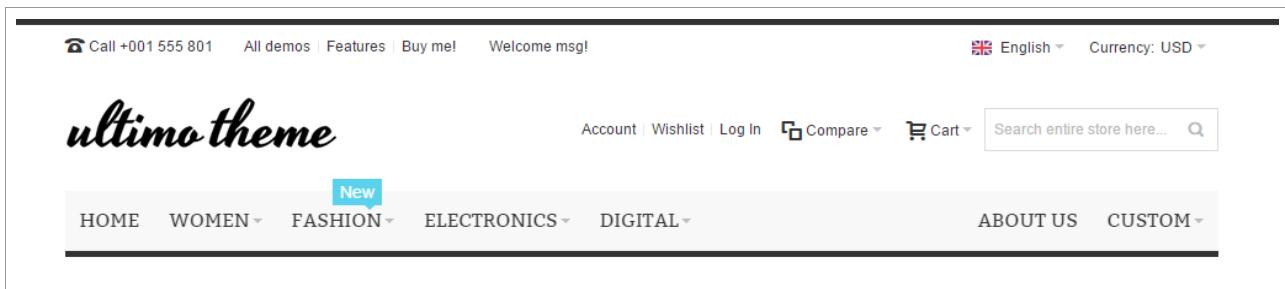
Now you need to assign the search box to the Right Column instead of the default Central Column. The Central Column is not displayed so any blocks assigned to the Central Column will also not be displayed.

Important:

if given header section is not displayed, then any blocks assigned to that section will also not be displayed.

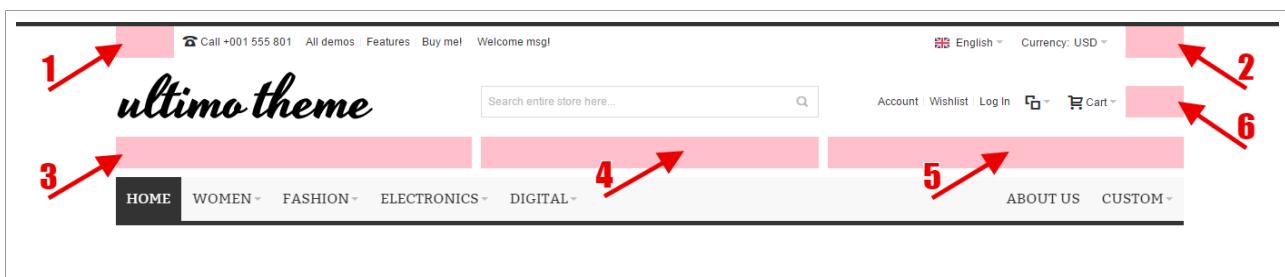
To assign search box to the Right Column, in the **Position** field under *System > Configuration > Theme Settings > Header > Search* select “Inside User Menu...” and then in the **Position Inside User Menu** field select where the search box should be

displayed. The final effect can look like on the following screenshot:



7.1.2 Header block positions

In section *System > Configuration > Theme Settings > Header* of admin panel you can select where to display some of the header blocks. Below you can see a list of all available positions:



1. Top, Left
2. Top, Right
3. Primary, Left Column
4. Primary, Central Column
5. Primary, Right Column
6. Inside User Menu

7.1.3 Header configuration options

In this chapter you can find a list of options available in the following section of the theme admin panel: *System > Configuration > Theme Settings > Header*.

Width of Primary Header Columns

- **Left Column** – width of the left section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.
- **Central Column** – width of the central section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.
- **Right Column** – width of the right section. Specify value in grid units (between 2 and 12). Important: Leave empty to remove this column.

Logo

- **Position** – select position of the site logo. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**.

Search

- **Position** – select position of the search box. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**. Select **Inside User Menu...** to show another option (see below).
- **Position Inside User Menu** – this field is visible only if **Inside User Menu...** was selected in the field above. Select position of the search box **Inside** the User Menu.

Account Links

This block is a container that groups Magento's Top Links (a set of links: *Account*, *Wishlist*, *Log In/Log Out*) and optional custom links. You can add custom links by creating a static block with identifier `block_header_top_links`. The content of this block will be displayed just next to the Top Links. Also, note that the Top Links can be disabled in theme settings, refer to **7.9 Default Magento blocks** for more details.

- **Position** – select position of the Account Links.

- **"Sign Up" Link** – if enabled, the "Sign Up" link will be displayed next to the "Log In" link.

User Menu

This block is a container that groups a few elements. By default it contains the following blocks: Mini Cart drop-down, Mini Compare drop-down, Account Links.

- **Position** – select position of the User Menu. Screenshot with all available positions can be found in this chapter: [7.1.2 Header block positions](#).
- **Position of Additional Line Break** – e.g. add a line break after the Mini Cart to force other elements of the User Menu to be displayed in the new line (below the Mini Cart). Note that elements of the User Menu are rendered in the following order: Mini Cart, Mini Compare, Top Links.

Main menu

- **Position** – select position of the main menu. Screenshot with all available positions can be found in this chapter: [7.1.2 Header block positions](#). If **Full Width Menu Container** was selected, menu will be displayed at the bottom of the header.

Mini Cart Drop-down

- **Position** – select position of the Mini Cart. Screenshot with all available positions can be found in this chapter: [7.1.2 Header block positions](#).
- **Show Text Label "Cart"** – if set to **No**, label "Cart" will be hidden in the Mini Cart drop-down block.

Note:

that in order to display Mini Cart drop-down block, option **Display Shopping Cart Sidebar** under *System > Configuration > Checkout > Shopping Cart Sidebar* has to be enabled in your Magento.

Mini Compare Drop-down

- **Display Mini Compare in Header** – enable/disable Mini Compare drop-down block in the header. If disabled, the block will be displayed on its default position in Magento (in the left sidebar).

- **Position** – select position of the Mini Compare. Screenshot with all available positions can be found in this chapter: **7.1.2 Header block positions**.
- **Show Text Label "Compare"** – if set to **No**, label "Compare" will be hidden in the Mini Compare drop-down block.

Currency Switcher

- **Position** – select position of the Currency Switcher. If you configure more than one currency in your Magento, the Currency Switcher switcher will appear automatically. You just need to configure currencies in your Magento and then clear Magento cache.

Language Switcher

- **Position** – select position of the Language Switcher.

The Language Switcher is just a store view switcher which is a standard feature of Magento. You can, for example, configure 3 store views (each with different language) in your Magento and the store view switcher will work as a language switcher.

Sticky Header

- **Sticky Header** – if enabled, the main menu will not disappear when the user scrolls down the page.
- **Full Width Sticky Header** – if set to **Yes**, content of the sticky header will be stretched to full width.
- **Show Logo in Sticky Header** – if set to **Yes**, additional logo image will be displayed in the sticky header (if the additional logo image was configured correctly). All details about configuration of the logo in Magento can be found in chapter **4.4 Logo**.

In order to display sticky logo:

1. Enable this option.
2. Upload your sticky logo image to the same folder where the main logo is stored.
3. The path of the main logo can be configured under **System > Configuration > Design > Header** in the **Logo Image Src** field. For

more details about configuration of the logo in Magento, refer to [4.4 Logo](#).

4. Name your sticky logo file the same as the file with the main logo, but add `_sticky` suffix to the file name.

For example:

if your main logo file is `my-logo.png`, then your sticky logo file has to be named like this: `my-logo_sticky.png`

5. Flush the cache.

Other Options

- **Move Switchers Next To Logo** – if set to **Yes**, language and currency switchers will be moved next to the logo (but only in mobile header) to make the header smaller.

7.2 Category View

In this tab, you can configure the options related to the category view. Category view presents the list of products from the selected category.

Images

- **Image Width** – maximum width of the product image. If width is not specified, default width will be used.

Remember that in the category view the visual size of the product image depends on the number of columns. For example, if you increase the number of columns, columns will be narrower and there will be less space for the image in each column, so the visual size of the image will be smaller than the actual dimensions of the image.

- **Keep Image Aspect Ratio** – set **Yes**, to keep the aspect ratio of the product images intact. Height of the images will be calculated automatically based on width.

Alternative Image

- **Alternative Image** – show alternative image on mouse hover over the main image.
- **Select Alternative Image By Column** – which column (attribute) of the product image gallery will mark the alternative image. Available columns:

- **Label**
- **Sort Order**

(See examples on the next page.)

- **Alternative Image Column Value** – specify the value of the column (attribute) which will mark alternative images of your products (see examples on screenshots below).

| Image | Label | Sort Order | Base Image [STORE VIEW] | Small Image [STORE VIEW] | Thumbnail [STORE VIEW] | Exclude | Remove |
|----------|--------------|------------|----------------------------------|----------------------------------|----------------------------------|--------------------------|--------------------------|
| No image | | 0 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | alt | 0 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | Sample label | 1 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | | 2 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | Other label | 3 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Example 1

| Image | Label | Sort Order | Base Image [STORE VIEW] | Small Image [STORE VIEW] | Thumbnail [STORE VIEW] | Exclude | Remove |
|----------|--------------|------------|----------------------------------|----------------------------------|----------------------------------|--------------------------|--------------------------|
| No image | | 0 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | alt | 0 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | Sample label | 1 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | | 2 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | Other label | 3 | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Example 2

Example 1:

If *Select Alternative Image By Column* is set to *Sort Order*, specify the number which will mark alternative images, e.g. 2.

Note:

in this example 2 doesn't mean second image, but the image which has value 2 in the *Sort Order* column in the image gallery.

Example 2:

If *Select Alternative Image By Column* is set to *Label*, specify label which will mark alternative images of your products. The label should be a short single word, e.g. alt.

7.3 Category View (Grid Mode)

In this tab, you can configure the options related to the **grid mode** of the category view.

Number of Columns

In the grid mode, products are displayed in columns. In this section of admin panel you can specify number of columns (products) in a row visible with a particular browser viewport width (breakpoint).

Below is the list of all available breakpoints for which you can specify the number of columns. Later in this chapter you can also find examples and explanation how exactly to use these fields.

- **Number of Columns Above 1920 px (breakpoint 3XL)** – number of columns displayed if browser viewport width is greater than 1920 px. The default value is 6.
- **Number of Columns Above 1680 px (breakpoint 2XL)** – number of columns displayed if browser viewport width is greater than 1680 px. The default value is 6.
- **Number of Columns Above 1440 px (breakpoint XL)** – number of columns displayed if browser viewport width is greater than 1440 px. The default value is 5.
- **Number of Columns Above 1200 px (breakpoint L)** – number of columns displayed if browser viewport width is greater than 1200 px. The default value is 4.
- **Number of Columns Above 992 px (breakpoint M)** – number of columns displayed if browser viewport width is greater than 992 px. The default value is 3.
- **Number of Columns Above 768 px (breakpoint S)** – number of columns displayed if browser viewport width is greater than 768 px. The default value is 3.
- **Number of Columns Above 640 px (Breakpoint XS)** – number of columns displayed if browser viewport width is greater than 640 px. The default value is 3.
- **Number of Columns Above 480 px (Breakpoint 2XS)** – number of columns displayed if browser viewport width is greater than 480 px. The default value is 2.
- **Number of Columns Above 320 px (Breakpoint 3XS)** – number of columns displayed if browser viewport width is greater than 320 px. Below 320 px products are automatically displayed in a single column. The default value is 2.

Important:

note that you can specify the number of columns **only** for breakpoints (viewport widths) which are **less than or equal** to the maximum breakpoint which is configured in the *Maximum Breakpoint* field (under *System > Configuration > Theme Layout*).

Refer to chapter **9.1 Responsive Layout** for more details about configuration of the maximum breakpoint.

Example 1:

If the *Maximum Breakpoint* field (which defines the maximum breakpoint for the entire store) is set to *1200 px*, then you can specify the number of columns in the following fields:

- *Number of Columns Above 1200 px (breakpoint L)*
- *Number of Columns Above 992 px (breakpoint M)*
- *Number of Columns Above 768 px (breakpoint S)*, and in the fields for all the lower breakpoints.

But you can't specify the number of columns in the *Number of Columns Above 1440 px (breakpoint XL)* field (or in the fields for larger breakpoints) because that breakpoint (*1440 px*) is **larger** than the value configured in the *Maximum Breakpoint* field (*1200 px*).

So on the frontend even when browser viewport width is greater than *1200 px* (which is the maximum breakpoint in this example), the grid will have as many columns as configured in the *Number of Columns Above 1200 px (breakpoint L)* field.

Example 2:

If the *Maximum Breakpoint* field (which defines the maximum breakpoint for the entire store) is set to *1680 px*, then you can specify the number of columns in the following fields:

- *Number of Columns Above 1680 px (breakpoint 2XL)*

- Number of Columns Above 1440 px (breakpoint XL)
- Number of Columns Above 1200 px (breakpoint L), and in the fields for all the lower breakpoints.

But you can't specify the number of columns in the Number of Columns Above 1920 px (breakpoint 3XL) field because that breakpoint (1920 px) is larger than the value configured in the Maximum Breakpoint field (1680 px).

Product Grid Display

- Equal Height of Grid Items (Products) – if enabled, all grid items will have the same height and "Add to cart" button will be displayed at the bottom of the item. This option is limited to product grid in category view and will not be applied to product sliders.
- Product Hover Effect – enable/disable effect which is visible on mouse hover over the product: block is emphasized with frame and shadow.

Note:

enable in order to use option "Display On Hover" in **Display Selected Elements**. Otherwise displaying elements on mouse hover over the product will not work properly.

- Disable Hover Effect Below... – product Hover Effect will be disabled if width of the browser viewport is below selected value.
- Hide "Add to" Links Below... – "Add to wishlist" and "Add to compare" links will be hidden if width of the browser viewport is below selected value.
- Align Center – align to the center elements of the grid item: product name, price, button etc.
- Size of Elements – control the size of the elements of the grid item, such as: product name, price, button.

Display Selected Elements

In this section, you can configure the display of selected elements of the grid items (products). Each of the elements can be: **a)** disabled and not displayed in the grid, **b)** displayed in a standard way, **c)** displayed only on mouse hover over the product.

Below is the list of the elements which can be configured:

- **Name** – name of the product
- **Display Name in Single Line** – If set to "Yes": if product name is too long to be displayed in a single line, it will be clipped and an ellipsis (... U+2026 HORIZONTAL ELLIPSIS) will be appended at the end of the line to represent the clipped text. This will only take effect if the field **Name** is set to "Display".
- **Price** – price of the product
- **Ratings** – product ratings (stars and number of reviews)
- **Add To Cart** – "Add to cart" button
- **"Add to" Links** – "Add to wishlist" and "Add to compare" links
- **Show "Add to" Links as Icons Above the Image** – if set to **No**, "Add to wishlist" and "Add to compare" links will be displayed as text links. Otherwise, links will be displayed as simple icons above the product image.

Note:

this option also affects "Add to wishlist" and "Add to compare" links in product sliders: **Featured Products Slider** and **New Products Slider**.

7.4 Category View (List Mode)

In this tab, you can configure the options related to the **list mode** of the category view.

- **Product Hover Effect** – enable/disable effect which is visible on mouse hover over the product: block is emphasized with frame and shadow.
- **Show "Add to" Links as Text Links** – If set to **No**, "Add to wishlist" and "Add to compare" links will be displayed as text links with icons.

7.5 Product Page

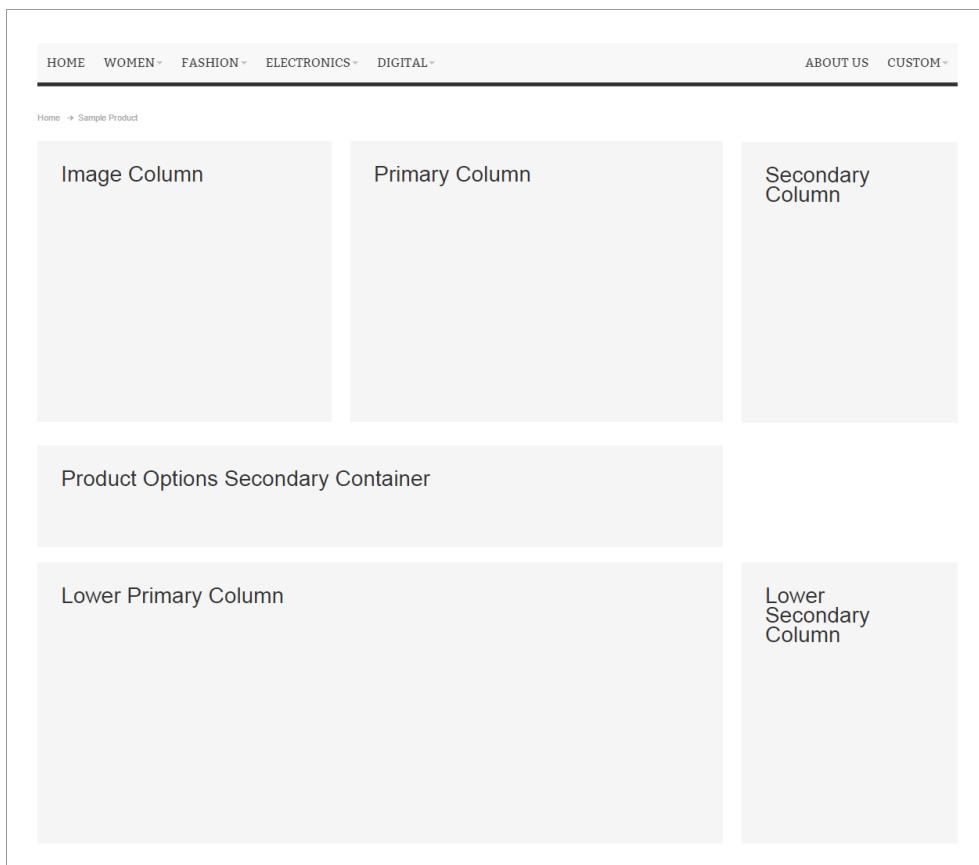
This chapter refers to the following section of the theme admin panel:
System > Configuration > Theme Settings > Product Page

List of all settings available in that section can be found later in this chapter: **7.5.3 Product page configuration options**.

7.5.1 Product page structure

Width of the product page sections is specified in grid units. As this is a 12-column grid, each section spans a number of those 12 columns, and should always add up to 12 for each row.

Below you can see all available sections:

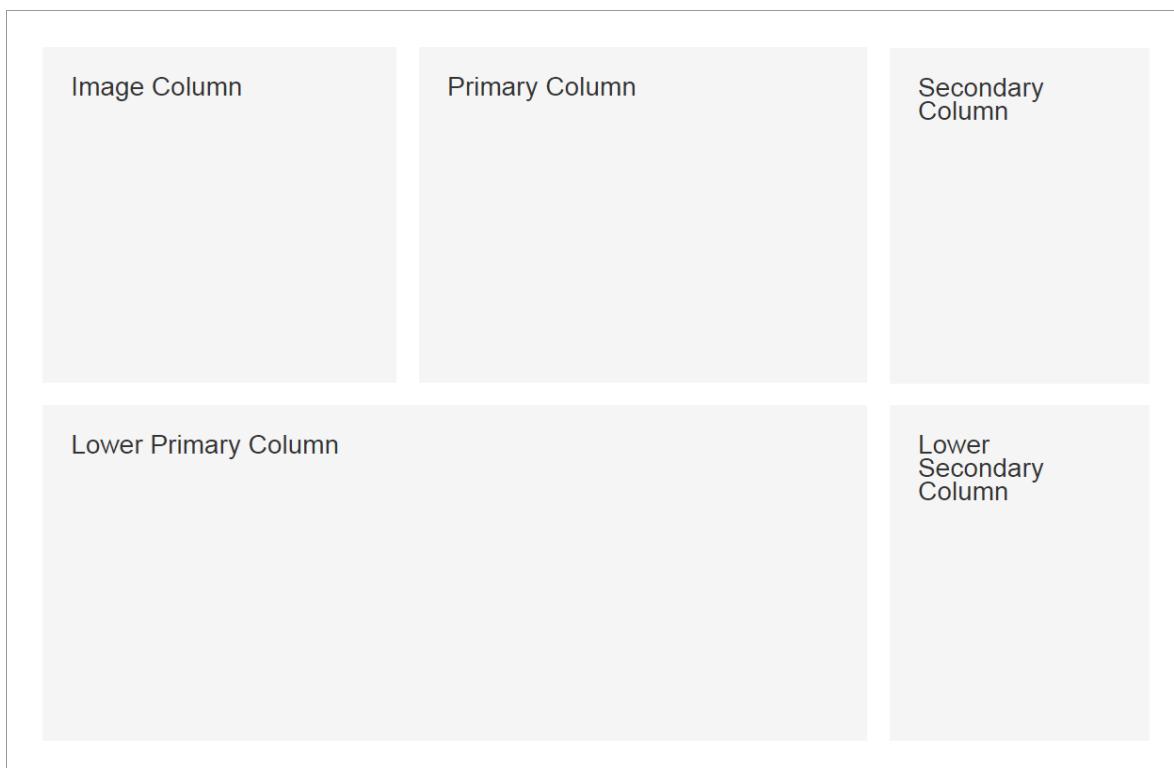


Example 1:

If you want to display three main sections (Image Column, Primary Column, Secondary Column) in a single row, sum of these three sections has to be equal **12 grid units**. For example:

```
Image Column = 4, Primary Column = 5, Secondary Column = 3  
4 + 5 + 3 = 12  
All the main three sections will be displayed in one row.  
  
Lower Primary Column = 9, Lower Secondary Column = 3  
9 + 3 = 12  
Lower sections will be displayed in another row, below the main sections.
```

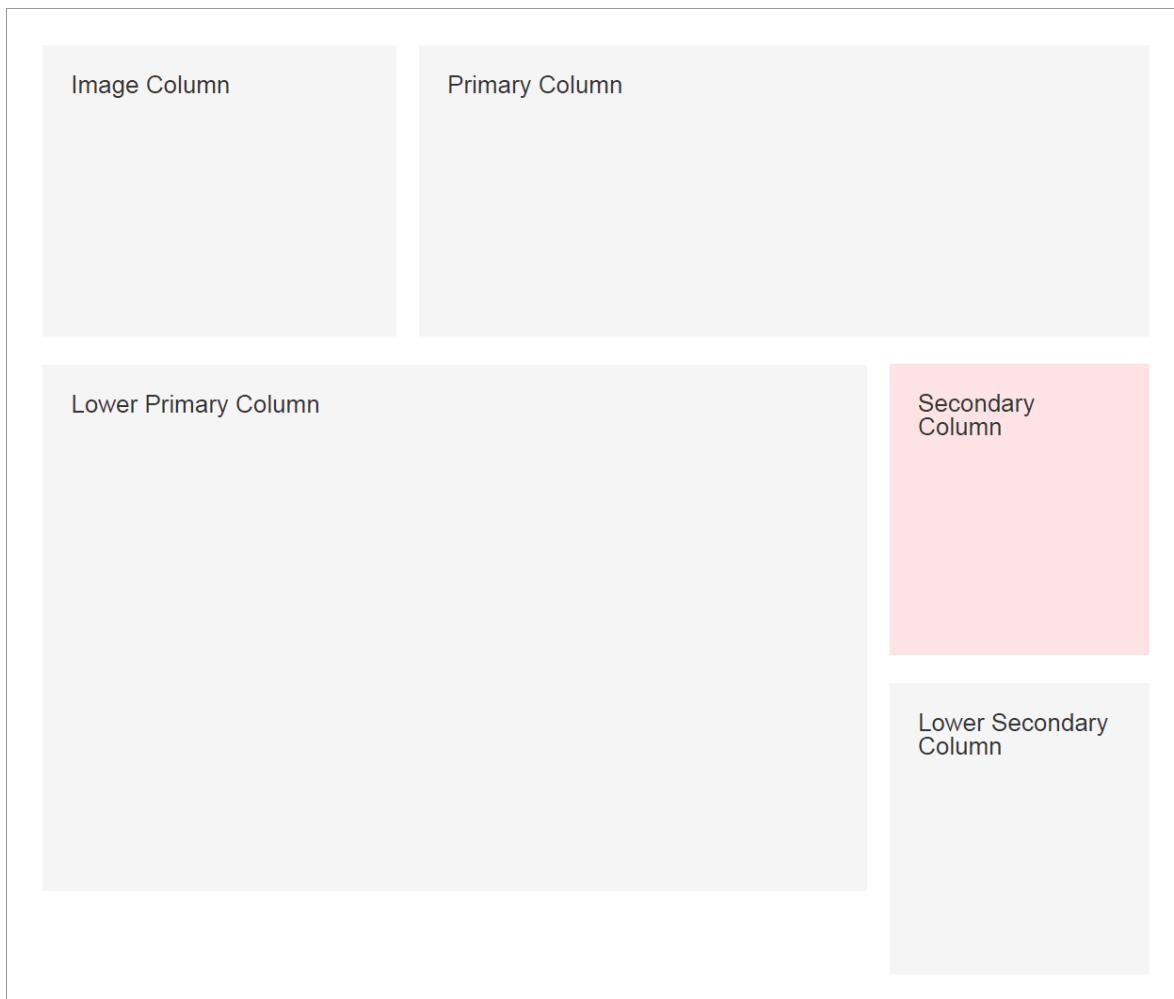
Note that the Product Options Secondary Container is optional (see more info later in this chapter) and will be displayed only if product options were assigned to "Block after Info Column" in product's settings. Otherwise, options container will not be displayed:



But if sum of these three sections is greater than 12, the last section will drop to the next row. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = 3  
4 + 8 + 3 = 15  
Image Column and Primary Column will be displayed in one row, and Secondary Column will drop to  
the next row.  
  
Lower Primary Column = 9, Lower Secondary Column = 3  
9 + 3 = 12
```

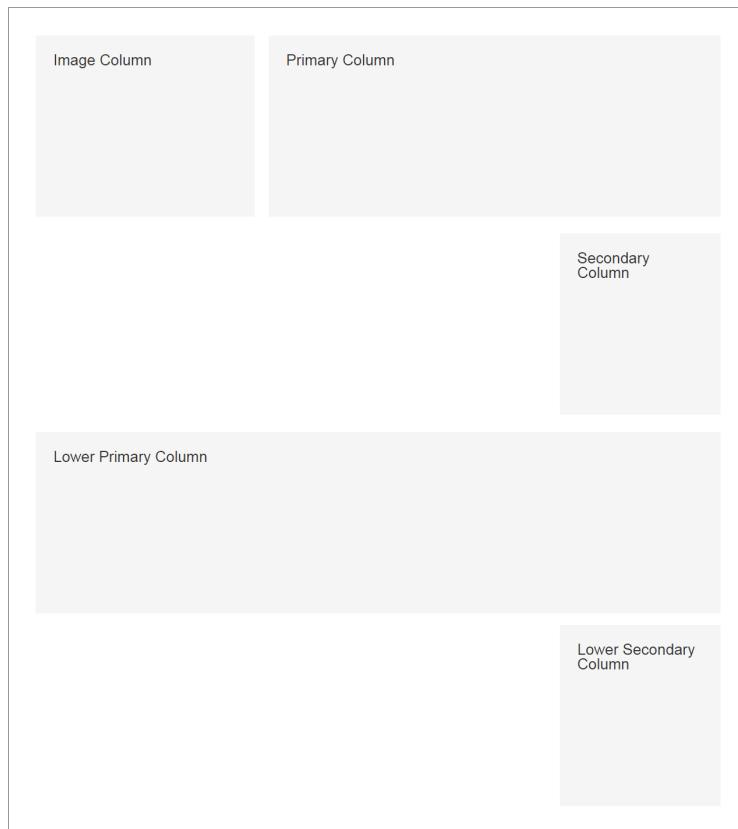
Note that the Secondary Column always **floats to the right side** so it will be placed above the Lower Secondary Column and at the side of the Lower Primary Column:



If the Lower Primary Column is wider, then it will be displayed in another row, below the Secondary Column. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = 3  
4 + 8 + 3 = 15  
Secondary Column will drop to the next row.  
  
Lower Primary Column = 12, Lower Secondary Column = 3  
12 + 3 = 15
```

The Lower Primary Column takes entire available space so it will drop to the next row, below the Secondary Column. And the Lower Secondary Column will drop to another row, below the Lower Primary Column.



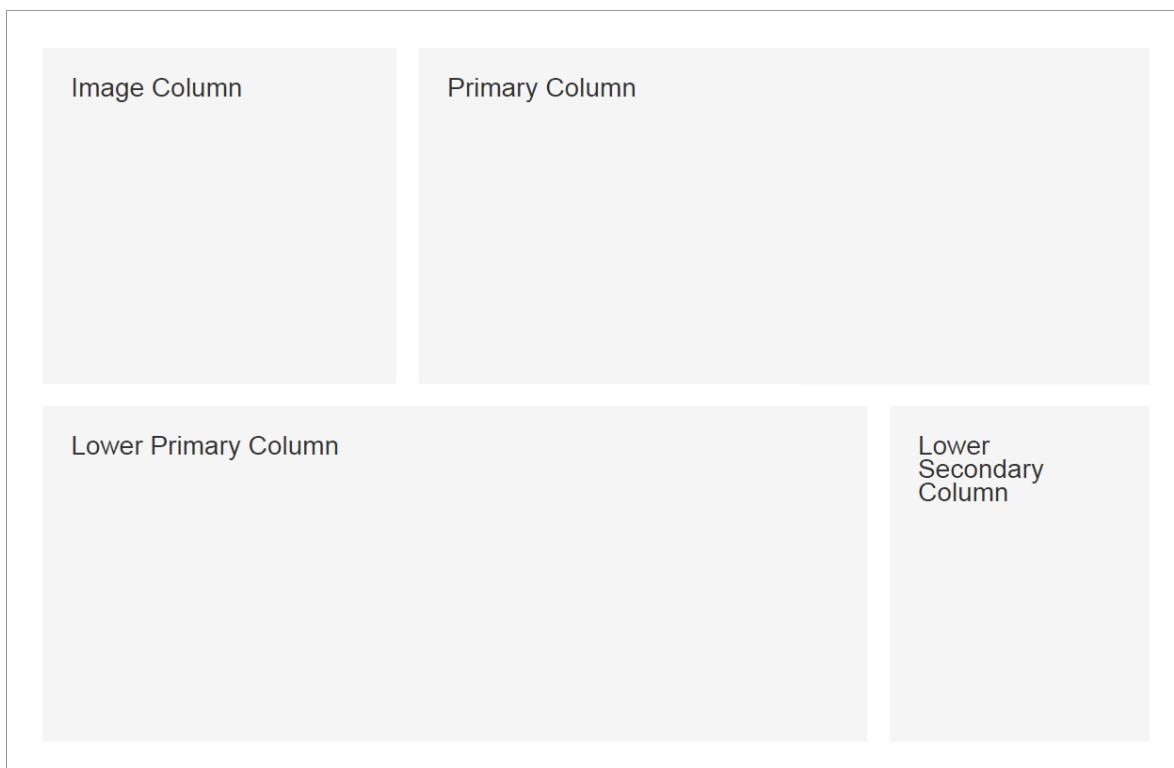
In this case it's recommended to make the Lower Primary Column narrower so that it fits the empty space at the left side of the Secondary Column.

Example 2:

If you want to get rid of the Secondary Column, leave the **Secondary Column Width** field empty and make the Primary Column wider. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = leave empty  
4 + 8 + ... = 12  
  
Lower Primary Column = 9, Lower Secondary Column = 3  
9 + 3 = 12
```

The Image Column and the Primary Column will take the entire row and the Secondary Column will disappear:

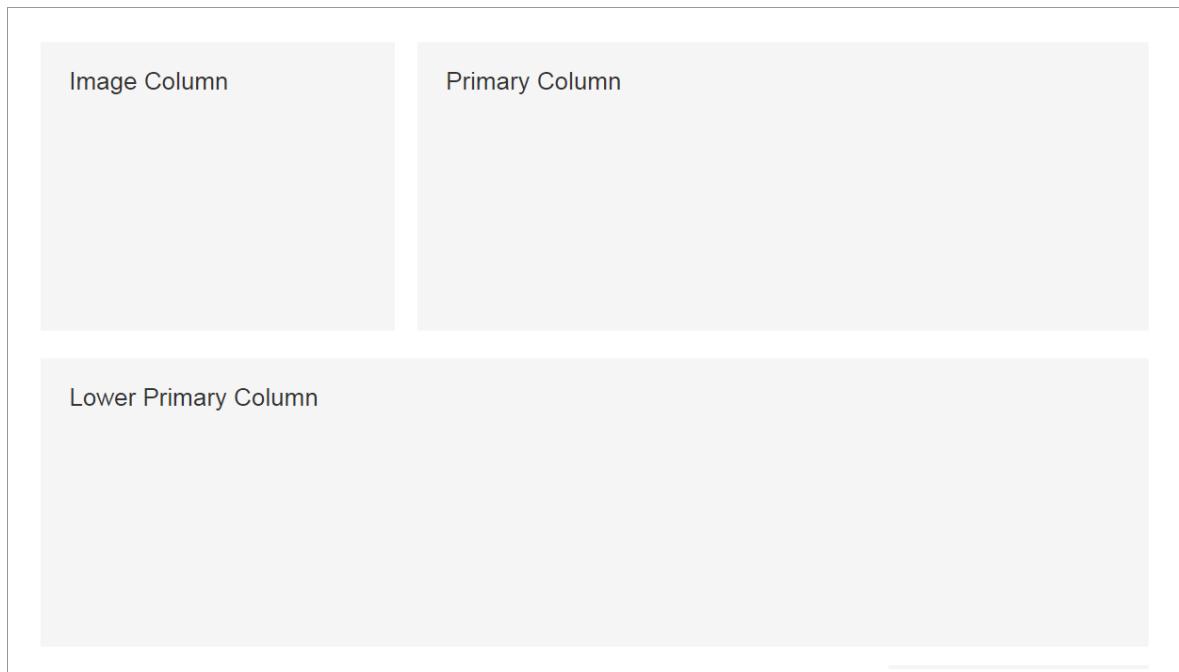


Example 3:

If you want to get rid of both: the Secondary Column and the Lower Secondary Column, leave the **Secondary Column Width** and the **Lower Secondary Column Width** fields empty and make the Primary Column and the Lower Primary Column wider. For example:

```
Image Column = 4, Primary Column = 8, Secondary Column = leave empty  
4 + 8 + ... = 12
```

```
Lower Primary Column = 12, Lower Secondary Column = leave empty  
12 + ... = 12
```



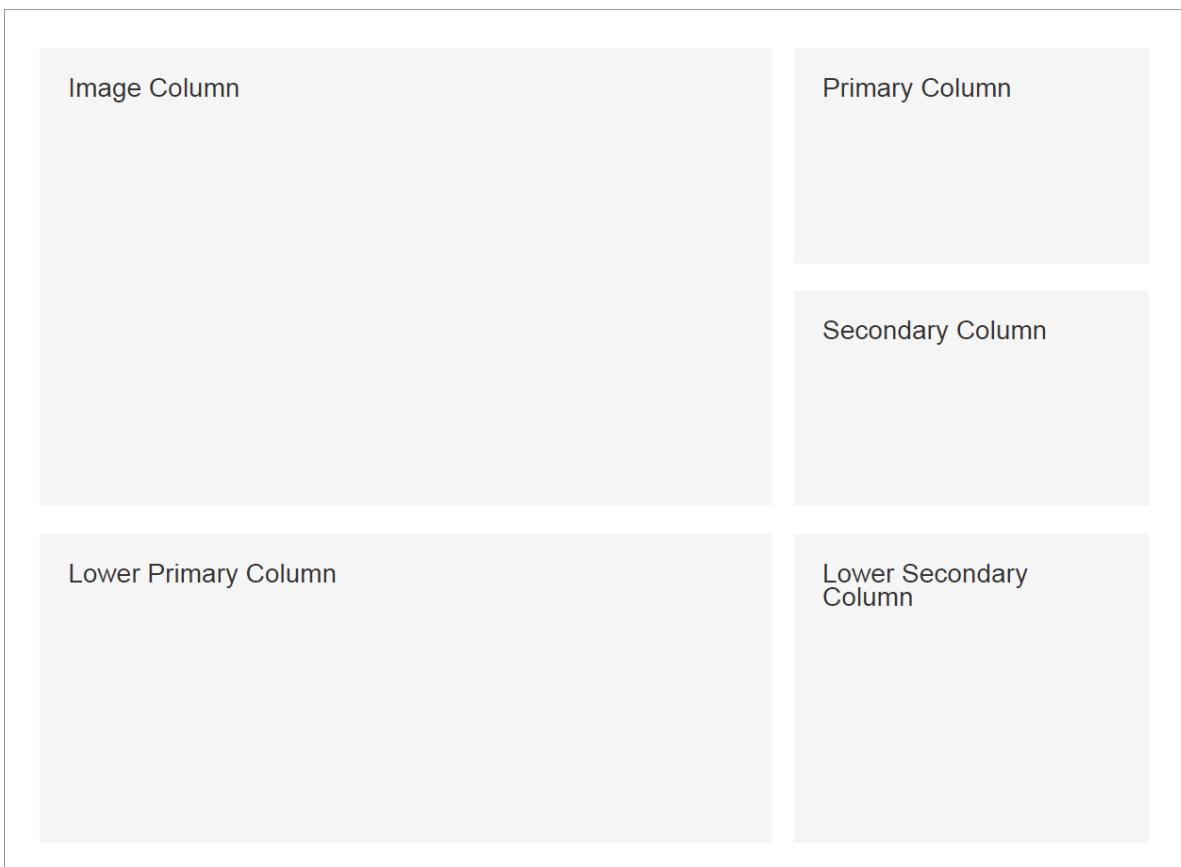
Example 4:

If sum of the Image Column and the Primary Column equals 12, and the Primary Column and the Secondary Column both have the same width, sections will have the following layout:

```
Image Column = 8, Primary Column = 4, Secondary Column = 4  
8 + 4 + 4 = 16
```

```
Lower Primary Column = 9, Lower Secondary Column = 3  
8 + 4 = 12
```

This allows you to present large product images. In this example, both the Primary Column and the Secondary Column will be displayed as a simple sidebar:



7.5.2 Product page block positions

In section *System > Configuration > Theme Settings > Product Page* of admin panel you can select where to display some of the product page blocks. Below you can see a list of all available positions:

The screenshot shows a product page for 'Sample Product'. Red numbered arrows point to specific areas:

- 1**: Points to the top banner area above the product title.
- 2**: Points to the bottom banner area below the product details.
- 3**: Points to the secondary column header area.
- 4**: Points to the secondary column content area.
- 5**: Points to the social sharing icons at the bottom of the page.
- 6**: Points to the tab navigation bar at the bottom of the page.
- 7**: Points to the bottom-most banner area at the very bottom of the page.
- 8**: Points to the footer area below the tabs.

1. Primary Column, Position 1

2. Primary Column, Position 2

3. Secondary Column, Position 1

4. Secondary Column, Position 2
5. Secondary Column, Position 3
6. Lower Primary Column, Position 1
7. Lower Primary Column, Position 2
8. Lower Secondary Column, Position 1

7.5.3 Product page configuration options

In this chapter you can find a list of options available in the following section of the theme admin panel: *System > Configuration > Theme Settings > Product Page*.

Width of Columns

- **Image Column Width** – width of the section with the product image. Specify value in grid units. By default it is equal to 4.

If you change the value of the **Image Column Width**, you should also change the width of the main product image. You can do it in *System > Configuration > Zoom > Image Size > Main Image Width*. Optimal width of the image depends on two values:

- the width specified in the **Image Column Width** field
- the maximum width of the page specified in **Maximum Breakpoint** field in *System > Configuration > Ultimo > Theme Layout* section of the admin panel.

For example, if your **Maximum Breakpoint** is equal to 1024 px, and the **Image Column Width** is 4 units, there is no need to set the image width larger than 287 pixels, because the visible size of the image displayed inside the Image Column will not exceed the size of the Image Column.

The following table specifies the optimal **Main Image Width** for different sizes of the Image Column:

| | Maximum Breakpoint | | | | |
|--------------------|--------------------|---------|---------|---------|---------|
| Image Column Width | 1024 px | 1280 px | 1360 px | 1440 px | 1680 px |
| 4 units | 287 | 363 | 395 | 421 | 466 |
| 5 units | 366 | 463 | 504 | 536 | 593 |
| 6 units | 446 | 563 | 612 | 651 | 720 |

- **Primary Column Width** – width of the section with the product name, price, options etc. Specify value in grid units. By default it is equal to 5.
- **Secondary Column Width** – width of the section with additional product data and custom content. Specify value in grid units. By default it is equal to 3.

Important:

leave empty to remove this column.

- **Lower Primary Column Width** – width of the section with tabs, reviews etc. Specify value in grid units. By default it is equal to 9.
- **Lower Secondary Column Width** – width of the section with additional custom content. Specify value in grid units. By default it is equal to 3.

Important:

leave empty to remove this column.

- **Product Options Secondary Container Width** – width of the section with product options container. Specify value in grid units. By default it is equal to 9. It's recommended to specify the same width as in the **Lower Primary Column Width** field.

Note that product options will be displayed in this section only if:

- product has any option
- the "Display Product Options In" field in product's configuration (under *Catalog > Manage Products*) was set to "Block after Info Column".

If the "Display Product Options In" field was set to "Product Info Column", product options will be displayed inside the Primary Column.

Collateral Data

Container for product collateral data. It can be displayed as tabs, accordion or stacked blocks.

- **Display Collateral Data as Tabs** – display Collateral Data as tabs or accordion. If set to "No", Collateral Data will be displayed as stacked blocks.
- **Mode** – available modes of the tabs:
 - **Tabs** – display Collateral Data in tabs.
 - **Tabs/Accordion** – display Collateral Data in tabs which turn into accordion if browser viewport is narrower than the width specified in the "Accordion Threshold" field.
 - **Accordion** – display Collateral Data in accordion.
- **Accordion Threshold** – if option "Mode" was set to "Tabs/Accordion", tabs will turn into accordion if browser viewport is narrower than the width specified in this field. Enter value in pixels. This option is available only if "Mode" was set to "Tabs/Accordion".
- **Accordion Initially Collapsed** – if set to "Yes", if accordion is visible on page load, all blocks of the accordion will be collapsed. This option is available only if accordion was enabled.
- **Small Size** – display small tabs/accordion. Useful when Collateral Data is displayed in a narrow section of the product page.
- **Position** – position of Collateral Data container. Screenshot with all available positions can be found in this chapter: **7.5.2 Product page block positions**.

Collateral Data - Selected Blocks

Select blocks which will be displayed in the Collateral Data container. Collateral Data can be displayed in different modes: tabs, accordion or stacked blocks. For more info, see the section above.

- **Reviews in Collateral Data** – display product reviews in Collateral Data container.

Important:

if position of the Collateral Data container is one of the following: Primary Column, Position 1 or Primary Column, Position 2 or Secondary Column, Position 1 or Secondary Column, Position 2 or Secondary Column, Position 3, reviews can't be included in the Collateral Data container.

- **Tags in Collateral Data** – display product tags in Collateral Data container.

Important:

if position of the Collateral Data container is one of the following: Primary Column, Position 1 or Primary Column, Position 2 or Secondary Column, Position 1 or Secondary Column, Position 2 or Secondary Column, Position 3, tags can't be included in the Collateral Data container.

- **Related Products in Collateral Data** – display related products in Collateral Data container.
- **Up-sell Products in Collateral Data** – display up-sell products in Collateral Data container.

Related Products

- **Position** – position of the Related Products block on product page. Screenshot with all available positions can be found in this chapter: **7.5.2 Product page block positions**.

Note:

this option **will be ignored** if option "Related Products in Collateral Data" was set to "Yes".

- **Related Products Layout** – layout of the Related Products block on product page:
 - **Simple Slider** – standard horizontal slider.

- **Thumbnails Slider** – in this case you can specify (in the "Number of Visible Products" field) the number of products displayed in a single slide.
- **Breakpoints** – products are displayed in columns – with this parameter you can specify the number of products in a row visible with a particular browser viewport width. Leave this field empty to use the default value.

Note:

this option is available only if the "Related Products Layout" field was set to "Simple Slider".

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers [A, B] number A indicates the web browser viewport width, number B indicates the number of products visible if browser viewport width is greater than the width specified in A.

In other words: display B products if viewport width is above A pixels.

- **Example 1 – the default value:**

```
[0, 1], [320, 2], [480, 3], [768, 1]
```

The default value is useful if the slider is displayed inside narrow sections, for example, inside the Secondary Column or Lower Secondary Column.

The default value can be translated as: display 1 product if viewport width is between 0px and 320px, display 2 products if viewport width is above 320px, display 3 products if viewport width is above 480px, and display 1 product if viewport width is above 768px.

This means that the wider is the screen, the more products will be visible in the slider, but **if the screen is wider than 768px** – only 1 product will be visible. Thanks to that, the slider can be displayed inside narrow columns.

- **Example 2:**

```
[0, 1], [320, 2], [480, 3], [960, 4], [1280, 5]
```

This example is useful if the slider is displayed inside wide section, for example: inside the Lower Primary Column.

This example can be translated as: display 1 product if viewport width is between 0px and 320px, display 2 products if viewport width is above 320px,

display 3 products if viewport width is above 480px, display 4 products if viewport width is above 960px, and display 5 products if viewport width is above 1280px.

- **Example 3:**

```
[0, 3]
```

which can be translated as: always display 3 products.

- **Number of Visible Products** – maximum number of related products displayed in a single slide. This field is visible only if "Thumbnails Slider" was selected in the "Related Products Layout" field.
- **Timeout** – to enable automatic scrolling of the slider, specify time (in milliseconds) between transitions. Leave empty to disable automatic scrolling. Note that 1000 miliseconds = 1 second.
- **Replace Related Products With Static Block** – **this option is deprecated and it will be removed**. Related Products can be replaced with a static block. To replace Related Products, first create a static block with ID `block_product_replace_related`.

Note:

this option **will be ignored** if option "Related Products in Collateral Data" was set to "Yes".

You can choose one of the following options:

- **Disable Completely** – disable Related Products block and don't replace it with any other content.
- **Don't Replace With Static Block** – never replace Related Products block with custom static block.
- **If Empty, Replace With Static Block** – replace Related Products block with custom static block only if the product does not have any related products.
- **Replace With Static Block** – always replace Related Products block with custom static block. It's useful when you want to disable Related Products block but still want to display some content in the same area.

Up-sell Products

- **Position** – position of Up-sell Products block on product page. Screenshot with all available positions can be found in this chapter: [7.5.2 Product page block positions](#).

Note:

this option **will be ignored** if option "Up-sell Products in Collateral Data" was set to "Yes".

- **Breakpoints** – with this parameter you can specify the number of products in a row visible with a particular browser viewport width.

Use the format presented in the examples below. Leave this field empty to use the default value.

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers [A, B] number A indicates the web browser viewport width, number B indicates the number of products visible if browser viewport width is greater than the width specified in A.

In other words: display B products if viewport width is above A pixels.

- **Example 1 – the default value:**

```
[0, 1], [320, 2], [480, 3], [960, 4], [1280, 5]
```

The default value is useful if the slider is displayed inside wide section, for example: inside the Lower Primary Column.

The default value can be translated as: display 1 product if viewport width is between 0px and 320px, display 2 products if viewport width is above 320px, display 3 products if viewport width is above 480px, display 4 products if viewport width is above 960px, and display 5 products if viewport width is above 1280px.

- **Example 2:**

```
[0, 1], [320, 2], [480, 3], [768, 1]
```

This example is useful if the slider is displayed inside narrow sections, for example, inside the Secondary Column or Lower Secondary Column.

This example can be translated as: display 1 product if viewport width is between 0px and 320px, display 2 products if viewport width is above 320px, display 3 products if viewport width is above 480px, and display 1 product if viewport width is above 768px.

This means that the wider is the screen, the more products will be visible in the slider, but **if the screen is wider than 768px** – only 1 product will be visible. Thanks to that, the slider can be displayed inside narrow columns.

- **Example 3:**

[0, 3]

which can be translated as: always display 3 products.

- **Timeout** – to enable automatic scrolling of the slider, specify time (in milliseconds) between transitions. Leave empty to disable automatic scrolling. Note that 1000 milliseconds = 1 second.
- **Replace Up-sell Products With Static Block** – **this option is deprecated and it will be removed.** Up-sell Products can be replaced with a static block. To replace Up-sell Products, first create a static block with ID `block_product_replace_upsell`.

Note:

this option **will be ignored** if option "Up-sell Products in Collateral Data" was set to "Yes".

You can choose one of the following options:

- **Disable Completely** – disable Up-sell Products block and don't replace it with any other content.
- **Don't Replace With Static Block** – never replace Up-sell Products block with custom static block.
- **If Empty, Replace With Static Block** – replace Up-sell Products block with custom static block only if the product does not have any up-sell products.
- **Replace With Static Block** – always replace Up-sell Products block with custom static block. It's useful when you want to disable Up-sell Products block but still want to display some content in the same area.

Other Elements

- **Show SKU** – display SKU on product page. SKU is an unique identifier of a product.

7.6 Product Labels

In this section of admin panel you can configure the options related to product labels.

Important:

in order to display product labels, you may need to reindex the data (**4.2 Index management**) and flush the cache (**4.1 Caching system**) after marking products as new and after setting a special price.

"New" Label

- **Show "New" Label** – if a product is marked as new, the "New" label will be displayed over the product.

Note:

To promote a product as new, go to *Catalog > Manage Products* and click selected product to go to the product edit page. Next, in *Set Product as New from Date* and *Set Product as New from Date* fields, enter the date range in which the product will be promoted as a new product.

"Sale" Label

- **Show "Sale" Label** – if a product is on sale, the "Sale" label will be displayed over the product.

Note:

To add the "Sale" label to a product, go to *Catalog > Manage Products*

and click selected product to go to the product edit page.

Next, enter a special price in the *Special Price* field.

In *Special Price From Date* and *Special Price To Date* fields you can also enter (it's optional) the date range in which the product will be automatically promoted with the special price. If you leave these fields empty, Magento will fill in the *Special Price From Date* field with the current date and the special price will apply until you remove it from the product.

- **Display Saved Amount** – if this option is enabled and if a product has a special price, the label will be displayed over the product to show the percentage which reflects the saved amount of money.

For example, if the regular price is \$100 and the special price is \$90, the label will show **-10%**.

This label has priority over the standard "Sale" label. This means that if the *Show "Sale" Label* field and the *Display Saved Amount* field are enabled and the product has a special price, only the label with saved amount will be displayed.

- **Precision of Saved Amount** – the saved amount of money will be rounded. Specify number of digits after the decimal point. Allowed values: between 0 and 4.

Style of Labels

- **Round Labels** – enable to change the shape of labels. If disabled, labels are rectangular.

7.7 Footer

In this tab, you can configure the options related to the footer section.

- **Automatic Column Width Calculation** – if enabled, width of the columns in the footer (two rows of static blocks) will be calculated automatically and columns will always have maximum available width – grid units will be applied to columns. Refer to [6.4 Grid system](#) for more details about grid units and grid system.

It will take effect on 2 rows of static blocks with the following identifiers:

- **first row:**
 - block_footer_column1
 - block_footer_column2
 - ...
 - block_footer_column6
- **second row:**
 - block_footer_row2_column1
 - block_footer_row2_column2
 - ...
 - block_footer_row2_column6

For example, if only 3 static blocks are enabled in the first row, those 3 blocks will be displayed as 3 equal columns.

If this option is disabled, then the width of each column will be equal to 2 grid units.

Note: this option will not be applied if exactly 5 blocks are enabled. This is due to limitations of the 12-column grid system – it isn't possible to display 5 equal columns, because 12 can't be divided by 5.

7.8 Product sliders

In this tab, you can configure the options related to product sliders.

Settings from this tab will be applied to all existing product sliders, also to Related Products and Ups-Sell Products on product page.

Note:

Timeout for *Related Products* and *Ups-Sell Products* can be specified individually, refer to [7.5 Product Page](#) for more details.

More info about product sliders can be found in [6.1.4 Featured Products Slider](#) and [6.1.5 New Products Slider](#).

- **Timeout** – to enable automatic scrolling of the product sliders, you need to specify the time (in milliseconds) between slide transitions. Leave it empty to disable automatic scrolling. Note that 1000 milliseconds = 1 second.
- **Speed** – duration (in milliseconds) of scrolling animation. Applied to scrolling with arrows (and touch swipe).
- **Automatic Scrolling Speed** – duration (in milliseconds) of scrolling animation. Applied to automatic scrolling (if **Timeout** was specified) and to scrolling with pagination.
- **Pause on Hover** – if enabled, when the slider is mouseovered then the automatic scrolling will pause.
- **Loop** – if enabled, scrolling will loop.
- **Lazy Loading** – delays loading of images. If enabled, images outside of viewport will not be loaded before user scrolls to them.

7.9 Default Magento blocks

In this tab, you can disable some of the default Magento blocks. If needed, some of these blocks can be then substituted by a corresponding static blocks. Refer to [6.2 Built-in CMS blocks](#) for more details about the static blocks provided with this theme.

- **Top Links** – links in the header section: *Account*, *Wishlist*, *Log In/Log Out*.
- **Footer Links** – links in the footer section: Site Map, Search Terms, Contact Us etc. This block can be substituted by the static block `block_footer_links`.
- **Newsletter Subscribe Form** – block with the newsletter subscription form located in the footer section. All visitors have the option to sign up by entering their email address. This block can be substituted by the static block `block_footer_primary_bottom_right`.

Enter the following code in the block content to display newsletter subscription form inside that block:

```
 {{block type="newsletter/subscribe" template="newsletter/subscribe.phtml"}}
```

By using this code, you can display newsletter subscription form in any other static block or on custom CMS page in the footer.

- **Store Switcher in the Footer** – store switcher displayed at the bottom of the page. It is only displayed if there is more than one active store in Magento. Note that this is something different than store view switcher. “Store” in this case is a group of store views. Refer to this article for more details:
[How Multiple Websites & Stores Work](#)
- **Product Page: Related Products Checkbox** – checkboxes displayed in the Related Products block (for products without additional options) on product page.

7.10 Disable Sidebar Blocks (On Home Page)

In this tab, you can remove some of Magento's default blocks **from the home page** sidebars.

7.11 Customization Settings

Settings in this tab are related to theme customization. For more information about theme customization refer to chapter [5.1 How to modify the theme](#).

- **Load Custom CSS** – if enabled, `custom.css` file (in which you can override the default styling of the theme) will be loaded in the frontend.

This might be useful when you want to make some changes in the theme design. All the CSS placed in this file **will not be lost** after upgrading the theme.

Note:

this option may be enabled/disabled individually for each store view.

To create `custom.css` you can use the file `SAMPLE.custom.css` which is located in `skin/frontend/ultimo/default/css/` directory. Simply create a copy of `SAMPLE.custom.css` and rename it to `custom.css`. Uncomment selected code or add your own to override the default styles of the theme.

In some situations custom CSS is not enough. When you need to modify theme's default template files, you should create a custom sub-theme. Refer to [5. Customization](#) for more details about custom sub-themes in Magento.

7.12 Data Import

This tab is responsible for the data import procedures after installation.

Theme Configuration Import/Export

Click to go to the import/export page **Import → Export →**

Import Static Blocks and Pages From Selected Demo

Import From Demo [STORE VIEW]

Import Static Blocks

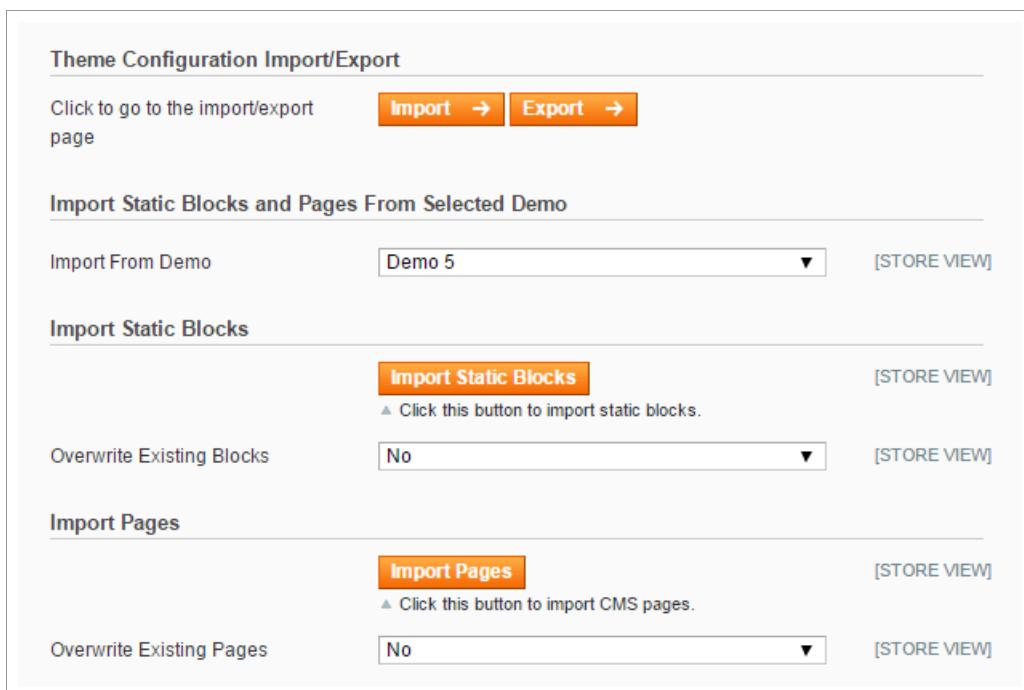
Import Static Blocks [STORE VIEW]
▲ Click this button to import static blocks.

Overwrite Existing Blocks [STORE VIEW]

Import Pages

Import Pages [STORE VIEW]
▲ Click this button to import CMS pages.

Overwrite Existing Pages [STORE VIEW]



The following functionalities are available:

- import and export theme configuration
- import sample content: static blocks and CMS pages

7.12.1 Theme Configuration Import/Export

To import/export theme configuration, go to the **Data Import** tab under *System > Configuration > Ultimo > Theme Settings* and click the appropriate button to display forms with additional settings:

The screenshot shows a section titled "Theme Configuration Import/Export". Below it is a link "Click to go to the import/export page". At the bottom are two orange buttons: "Import" and "Export".

Both options are described below.

Import

Click the *Import* button – you will be redirected to the configuration import form:

The screenshot shows the "Import Configuration" form. It has a "Import settings" section with two dropdown menus: "Select Configuration to Import" (set to "Demo2") and "Configuration Scope" (set to "Default Config"). A note below explains that imported settings will be applied to the selected scope. At the top right are "Reset Form" and "Import Configuration" buttons.

Using that form you can quickly import theme configuration from selected demo store. It will import theme settings to these sections in admin panel:

- *System > Configuration > Ultimo*
- *System > Configuration > Infortis Extensions*

After importing you can go to these sections and adjust the configuration to your needs. You can import as many times as you want, so the import functionality can be used to reset the configuration to default values.

To import, follow the instructions below:

1. On the form, choose one of the available demos from the *Select Configuration to Import* list.
2. From the *Configuration Scope* list select configuration scope – imported configuration settings will be saved in the selected scope.

Note:

it is recommended to import to the "Default Config" scope. Especially if you have only one store view in your Magento or if want all your store views to look similar. Settings from the "Default Config" scope can be later overridden on store view level, if needed.

Important:

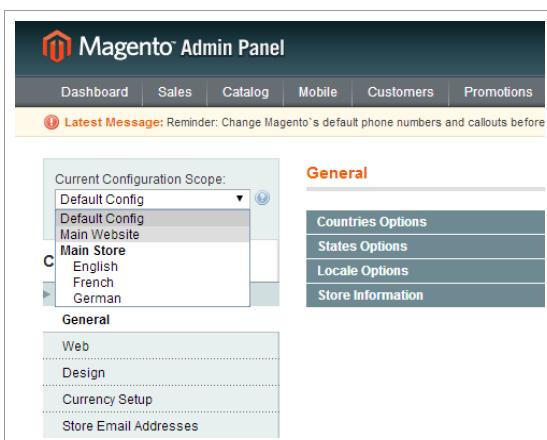
note that the scope of Magento configuration has three levels:

1. *"Default Config"*
2. *Website*
3. *Store view*

A lower level can override a higher level:

- settings of a store view override settings of a website
- and settings of a website override settings of "Default Config"

3. If you import to the *Default Config* scope, imported settings will be applied to all existing store views in your Magento. If you're not sure what is "scope" in Magento system configuration, please refer to the following page:
<http://www.magentocommerce.com/knowledge-base/entry/understanding-store-scopes>
4. Finally, click the *Import Configuration* button. Then, flush Magento cache (**4.1 Caching system**) in order to see changes.
5. After import, if you want to change the imported configuration, all the settings can be found in section *System > Configuration*. Before making changes in the configuration, select the scope using the *Current Configuration Scope* switcher which can be found in the top left corner – see the screenshot below:



For example, if you imported to the "Default Config", you need to select that scope using the *Current Configuration Scope* switcher. **Otherwise your changes will not take any effect.**

Export

Click the *Export* button – you will be redirected to the configuration export form:

The screenshot shows the 'Export Configuration' form. At the top right are 'Reset Form' and 'Export Configuration' buttons. The main area has a title 'Export settings'. It includes a 'File Name *' field with 'my-configuration' entered, a note explaining it's the name of the file, and a dropdown menu for 'Select Elements of the Configuration to Export' containing 'Ultimo', 'Brands', 'Zoom', 'Menu', and 'Slideshow'. Below that is a 'Configuration Scope *' dropdown set to 'All Store Views' with a note about saving the configuration for selected stores. A 'Help' icon is also present.

Using that form you can quickly export theme configuration to a file. It will export theme settings which can be found in the following sections of admin panel:

- *System > Configuration > Ultimo*
- *System > Configuration > Infortis Extensions*

On the *Select Elements of the Configuration to Export* list you can choose which sections of the configuration will be exported.

Specify the name of the file in which configuration will be saved. This name will appear on the configuration import form (which was described earlier in this chapter) so you will be able to import your configuration any time you want.

Exported configuration will be saved in a file in the following directory:
app/code/local/Infortis/Ultimo/etc/importexport/config/

7.12.2 Import Static Blocks and Pages From Selected Demo

Ultimo theme ships with a number of useful CMS pages and static blocks. To import these items into your system, navigate to the **Data Import** tab under *System > Configuration > Ultimo > Theme Settings*.

Import Static Blocks and Pages From Selected Demo

Import Static Blocks and Pages From Selected Demo

Import From Demo: Demo 5 [STORE VIEW]

Import Static Blocks

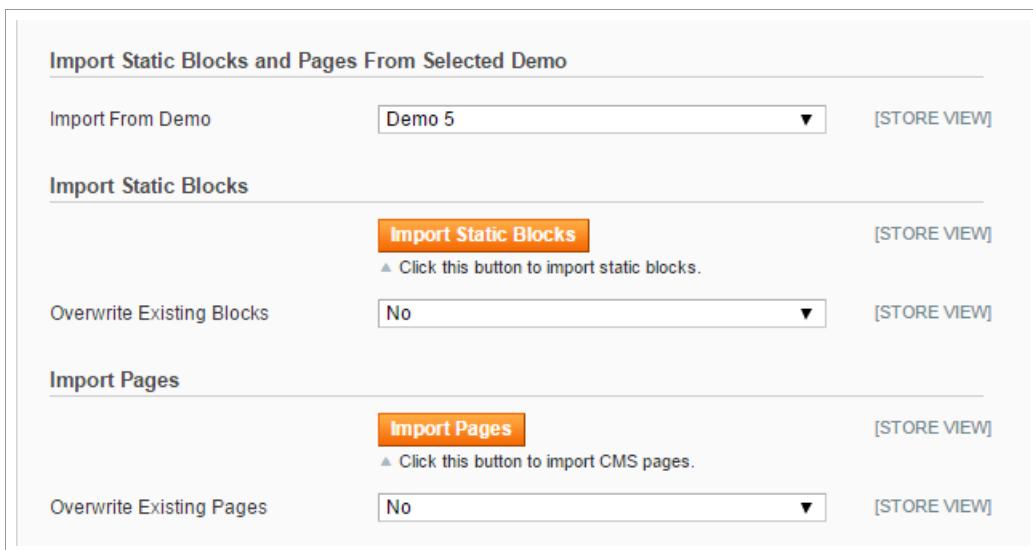
Import Static Blocks [STORE VIEW]
▲ Click this button to import static blocks.

Overwrite Existing Blocks: No [STORE VIEW]

Import Pages

Import Pages [STORE VIEW]
▲ Click this button to import CMS pages.

Overwrite Existing Pages: No [STORE VIEW]



Before importing, choose the demo from which you want to import blocks and pages.

- **Import From Demo** – from this dropdown list choose the demo from which you want to import blocks and pages, and immediately save the configuration by clicking the *Save Config* button. Save it before you import. Without saving, the *Import From Demo* option will remain unchanged after you refresh the admin page.

Import Static Blocks

- **Import Static Blocks** – click this button to import all predefined static blocks provided with this theme. Refer to **6.2 Built-in CMS blocks** for more details about the static blocks. Imported blocks will appear under *CMS > Blocks*.

To reset (re-import) selected static blocks, delete blocks which you want to reset, and then import blocks again. This way you will only import blocks which are missing.

Important:

in case you need the original content of some static blocks, XML file with all static blocks can be found in the following directory:
app/code/local/Infortis/Ultimo/etc/importexport/cms/

- **Overwrite Existing Blocks** – if set to **Yes**, imported blocks will overwrite existing blocks with the same identifiers (if any blocks with the same identifiers already exist). You can use this option when you want to restore the original content of all blocks.

Note:

to change this option select **Yes** or **No** and then **save the configuration**. Without saving, the option will remain unchanged after you refresh the page.

Import Pages

- **Import Pages** – click this button to import sample CMS pages provided with this theme. Imported pages will appear under *CMS > Pages*.

Sample home pages have identifiers such as `ultimo-home-page-01`. Home page configuration is described in this chapter: **6.1 Home page and CMS pages**

To reset (re-import) selected pages, delete pages which you want to reset, and then import pages again. This way you will only import pages which are missing (which were deleted).

Important:

in case you need the original content of CMS pages, XML file with all pages can be found in the following directory:
app/code/local/Infortis/Ultimo/etc/importexport/cms/

- **Overwrite Existing Pages** – If set to **Yes**, created pages will overwrite existing pages with the same identifiers (if you have any). You can use this option when you want to restore the original content of all pages.

Note:

to change this option select **Yes** or **No** and then **save the configuration**. Without saving, the option will remain unchanged after you refresh the page.

7.13 Integrations

This tab is responsible for integrations with selected third-party libraries or frameworks.

Bootstrap

- **Grid System** – enable/disable the grid system from Bootstrap (version 3.3.7).

Do not disable this option. Disable it only if you have custom modifications (or extensions) which already add the entire Bootstrap (including the grid system) to the frontend of your store. In that case you may want to disable this option in Ultimo theme.

Font Awesome

- **Font Awesome** – enable/disable Font Awesome, an icon toolkit based on CSS (version 4.7.0).

Chapter **6.5.3.E Font Awesome icons** explains the usage of Font Awesome icons.

8. Theme Design

This chapter refers to the following section of the theme admin panel:

System > Configuration > Ultimo > Theme Design

Ultimo is equipped with administrative module which gives you the ability to configure the visual appearance of your store. You can change the colors of many elements by using color pickers or by specifying the color code manually. You can also apply textures (patterns) for the main sections of the page.

Names of the most of the options are pretty self-explanatory. Some of them are described in this chapter.

Note:

disable Magento cache before you start configuring your store. Otherwise you will need to flush the cache in order to see any changes you have made in the config.

Important:

after creating new store view (or after changing the store view code), re-save theme settings (clicking **Save Config** button) in **System > Configuration > Ultimo > Theme Design** and in **System > Configuration > Ultimo > Theme Layout**. Otherwise design/layout settings will not be applied to that new store view.

8.1 Colors

In this tab, you can configure basic colors for the entire page. These colors will be inherited by all sections of the page.

Basic Colors

Text and links.

Buttons

Buttons such as “Add to cart”, “Log In” etc.

Interface Icons

Small buttons with icons, e.g.: arrows in product sliders, “sort order” in category view etc.

Custom Icons

Customizable icons (e.g. social icons) which can be used to build custom content.

Horizontal Links

Inline links, e.g. Magento's default Top Links in the header (set of links: Account, Wishlist, Log In), Footer Links etc.

Product Labels

Product labels “New” and “Sale”.

Other

Other elements not classified elsewhere.

8.2 Font

In this tab, you can configure the options related to fonts.

8.3 Effects

In this tab, you can set additional effects for some of the page elements.

- **Content Area Shadow** – If set to **Yes**, all sections with nontransparent **Inner Background Color**, will drop soft shadow.
For example, if you set any color in **Inner Background Color** field in **Footer – Primary** tab, shadow will be displayed in the main area of the footer section.
- **Main Menu Shadow** – select which container of the menu will drop shadow. This field will be visible if **Content Area Shadow** was enabled.

Note:

unlike other sections, menu doesn't need to have **Inner Background Color** to drop shadow.

- **Structural Elements Transition** – if enabled, main structural elements of the page layout will be animated during web browser window resizing. It is recommended to leave this option disabled for browser compatibility.

8.4 Page

In this tab, you can configure the options related to the background of the entire page.

- **Background Color** – background color of the entire page.
- **Background Image** – here you can upload an image which will be displayed as a background of the entire page. Click the **Browse...** button and locate the image on your computer.

Note:

sample backgrounds can be found in
media/wysiwyg/infortis/ultimo/custom/background directory.

Once configuration has been saved, a small icon of the existing image will appear to the left of the button. You can then configure the way the image is displayed with the following options:

- **Background Repeat** – defines how the background image is repeated. It can be repeated along the vertical axis, the horizontal axis, both, or not repeated at all. By default it is repeated both vertically and horizontally.
- **Background Attachment** – determines whether the background image is fixed or scrolls with the rest of the page. By default it scrolls.
- **Background Position (x-axis)** – defines the initial position of the background image along the horizontal axis. By default it is aligned to the center of the page.
- **Background Position (y-axis)** – defines the initial position of the background image along the vertical axis. By default it is aligned to the top of the page.
- **Pattern** – background pattern of the entire page. Pattern covers the background image.

8.5 Header

In this tab, you can configure the options related to the header section of the page. Background image can be set in exactly the same way as for entire page (which is described in previous chapter).

8.6 Header – Top

In this tab, you can configure the area at the top of the header. Settings from this tab will override settings from the *Header* tab.

8.7 Header – Primary

In this tab, you can configure elements of the primary area of the header.

8.8 Main Menu

In this tab, you can configure the options related to the main menu bar.

8.9 Main

In this tab, you can configure the options related to the main section of the page (the section located between the header and the footer).

8.10 Category View

In this tab, you can configure selected elements of the category view.

8.11 Product Page

In this tab, you can configure the product page (in Magento also called "product view").

8.12 Slideshow

In this tab, you can configure elements related to slideshows.

8.13 Footer

In this tab, you can configure the entire footer section of the page. Settings from this tab are inherited by all the other sub-sections of the footer (*Footer-Top*, *Footer-Primary* and *Footer-Bottom*). Selected settings can be overridden in sub-sections.

8.14 Footer – Top

In this tab, you can configure the options related to the area at the top of the footer. Settings from this tab will override settings from the *Footer* tab.

8.15 Footer – Primary

In this tab, you can configure the options related to the primary area of the footer. Settings from this tab will override settings from the *Footer* tab.

8.16 Footer – Secondary

In this tab, you can configure the options related to the secondary area of the footer (just below the primary area). Settings from this tab will override settings from the *Footer* tab.

8.17 Footer – Bottom

In this tab, you can configure the options related to the bottom area of the footer. Settings from this tab will override settings from the *Footer* tab.

9. Theme Layout

This chapter refers to the following section of the theme admin panel:

System > Configuration > Ultimo > Theme Layout

Note:

disable Magento cache before you start to configure your store. Otherwise you will need to flush the cache in order to see any changes you have made in the config.

Important:

after creating new store view (or after changing the store view code), re-save theme settings (clicking **Save Config** button) in **System > Configuration > Ultimo > Theme Design** and in **System > Configuration > Ultimo > Theme Layout**. Otherwise design/layout settings will not be applied to that new store view.

9.1 Responsive Layout

In this tab, you can configure the options related to the responsive layout.

- **Responsive Layout** – enable or disable responsive layout
- **Maximum breakpoint** – specify maximum breakpoint of the responsive layout. Maximum breakpoint also defines the **maximum width of the content area** of the page.

Content area

The content area is the region at the center of the viewport where the content is displayed. It is usually narrower than the viewport itself (unless you set the *Maximum Breakpoint* to *Full width* – then the content area will be stretched to cover the entire width of the viewport).

Note that each of the predefined breakpoints in the *Maximum Breakpoint* field corresponds with a value of the maximum width of the content area. But the actual width of the **content area is always a little bit narrower** than the selected value – the design just needs some additional space at the sides so that the content doesn't touch the edges of the viewport.

Maximum width of the content area

Here's the table which presents the maximum width of the content area depending on the maximum breakpoint:

| Maximum Breakpoint | Maximum width of the content area |
|--------------------|-----------------------------------|
| 1920 | 1740 |
| 1680 | 1500 |
| 1440 | 1380 |
| 1200 | 1170 |
| 992 | 970 |
| Full width | width of the viewport |

Actual width of the content area

The actual width of the content area depends on the value of the **side padding**

which can be configured by the admin in the *Side Padding* field under *System > Configuration > Theme Design > Page > Inner Container* (the side padding can also be configured individually for each section of the page, e.g. under *System > Configuration > Theme Design > Header-Top > Inner Container*).

The side padding increases the empty space at the sides of the content area. The default value of the side padding is `15 px` (at each side).

Here's the table which presents the actual width of the content area depending on the maximum breakpoint and the default value of the side padding:

| Maximum Breakpoint | Actual width of the content area |
|--------------------|---------------------------------------|
| 1920 | 1710 |
| 1680 | 1470 |
| 1440 | 1350 |
| 1200 | 1140 |
| 992 | 940 |
| Full width | width of the viewport minus 30 pixels |

To specify a custom width you can select `Custom width...` from the dropdown list (a new field will appear below).

- **Custom Maximum Breakpoint** – this field defines custom maximum breakpoint and custom maximum width of the content area. To display this field you need to select `Custom width...` from the *Maximum Breakpoint* field.

Specify value in pixels, do not add "px" suffix.

- **Fluid Width** – if this option is enabled, when the web browser window is resized, the width of the content area is fluid.

If this option is disabled, when the web browser window is resized, the width of the content area is static and changes only on layout breakpoints.

Note:

note that this option does not stretch the content area to full width (to cover the entire width of the viewport). To do that you need to set the *Maximum Breakpoint* field to `Full width`.

9.2 Disable responsive layout

In order to disable responsive layout, go to *System > Configuration > Ultimo > Theme Layout* section and set *Responsive Layout* field to *Disable*.

Some additional steps are also required:

- disable mobile version of the main menu: *Mode* under *System > Configuration > Top Menu*.
- do not use *is_responsive* parameter in sliders: **6.1.4 Featured Products Slider**, **6.1.5 New Products Slider**, **12.2.4 Brand Slider**. With this parameter sliders are responsive and change the number of visible items depending on the width of the web browser's viewport
- remove collapsible elements from the footer static blocks (e.g. *block_footer_column1*), they are not needed if the layout is not responsive. List of footer static blocks can be found in **6.2 Built-in CMS blocks**.

CSS classes responsible for collapsible elements are described in **6.5.2 Collapsible blocks**. Basically you need to remove classes *collapsible* and *mobile-collapsible* from the content of the static blocks or add *active* class to the element's wrapper so that the element is not collapsed and hidden by default anymore.

10. Zoom

This chapter refers to the following section of the theme admin panel:

System > Configuration > Zoom

It describes the Cloud Zoom (which is an implementation of a free jQuery plugin Cloud Zoom <http://www.professorcloud.com/>) and Lightbox on product page (which is an implementation of a free jQuery plugin Colorbox <http://www.jackmoore.com/colorbox/>).

10.1 Zoom configuration

Below you can find a list of the Zoom settings available in the admin panel.

10.1.1 General

- **Enable Module** – this option should be enabled in order to use zoom features on product pages. If disabled, images will be displayed in the same way as in Magento's "default" theme.
- **Use Cloud Zoom** – If set to **Yes**, Cloud Zoom will be used to zoom product images on product pages.
- **Position** – specify the position of the zoom area relative to the main image. If **Inside** is selected, zoom area will be placed inside the main image – useful if you would rather not obscure any other content.
- **Lens Opacity (%)** – specify opacity of the lens pointer.
- **Zoom Area Width** – width of the zoom area. Leave empty to use default (the width will be the same as the main image).

Note:

Field **Zoom Area Width** and **Zoom Area Height** will be ignored if **Position** is set to "Inside".

- **Zoom Area Height** – height of the zoom area. Leave empty to use default (the height will be the same as the main image).
- **Image Width After Zooming** – width of the enlarged image. This field refers to the actual resolution of the image after zooming.
- **Image Height After Zooming** – height of the enlarged image. This field refers to the actual resolution of the image after zooming.
- **Tint Color** – color which will cover the main image. Specify color in hexadecimal format, e.g. #12bbcc (not more than 7 characters). Leave empty to disable this effect. Does not work if **Soft Focus** is enabled.
- **Tint Opacity (%)** – specify opacity of the tint, where 0 is fully transparent, and 1 is fully opaque.
- **Soft Focus** – applies a subtle blur effect to the small image. Does not work if **Tint Color** effect is enabled.
- **Smooth Move** – Amount of smoothness/drift of the zoom image as it moves. The higher the number, the smoother/more drifty the movement will be. 1 = no smoothing.

10.1.2 Image Size

In this tab, you can override the default size of the product images.

- **Keep Image Aspect Ratio** – set **Yes**, to keep the aspect ratio of the product images on product page.

Note:

If set to **Yes**, height specified in **Main Image Height** and **Image Height After Zooming** will be ignored and the actual height of the images will be calculated automatically based on width.

- **Main Image Width** – specify width of the main image on product page.

Note:

Remember that uploading images smaller than the size specified in **Main Image Width** and **Main Image Height** fields may impact the quality of the images displayed on product page.

- **Main Image Height** – specify the height of the main image on product page. Leave empty to keep the aspect ratio of the image intact (**height will be calculated automatically based on width**). If the width is not specified, default width and height will be used.

10.1.3 Thumbnails

In this tab, you can enable a slider for product image thumbnails.

- **Use Slider** – enable/disable slider functionality for product image thumbnails.
- **Timeout** – to enable automatic scrolling of the slider, you need to specify the time (in milliseconds) between slide transitions. Leave it empty to disable automatic scrolling. Note that 1000 milliseconds = 1 second.
- **Speed** – duration (in milliseconds) of scrolling animation. Applied to scrolling with arrows (and touch swipe).
- **Automatic Scrolling Speed** – duration (in milliseconds) of scrolling animation. Applied to automatic scrolling (if **Timeout** was specified).
- **Loop** – if enabled, scrolling will loop.
- **Move Items** – number of thumbnails that should move on animation. Allowed values:
 - 1 – will move one thumbnail.
 - 0 – will move all visible thumbnails.
- **Lazy Loading** – delays loading of images. If enabled, images outside of viewport will not be loaded before user scrolls to them.

10.1.4 Lightbox

- **Enable Lightbox** – if enabled, lightbox functionality will be available for the main product image.
 - If the **Use Cloud Zoom** field is set to **Yes**, zoom button will be displayed above the main product image. After clicking the zoom button, the image will be displayed in the lightbox
 - If the **Use Cloud Zoom** field is set to **No**, the entire product image can be clicked to display the image in the lightbox
- **Maximum Width** – maximum width of the lightbox content. Leave empty to show image in its maximum size (specified in **Image Width After Zooming** field). You can enter values in the following formats:
 - 90%
 - 500px
 - 500
- **Maximum Height** – maximum height of the lightbox content. Leave empty to show image in its maximum size (specified in **Image Height After Zooming** field). Available formats: see the field above.
- **Lightbox As Gallery** – if set to **Yes**, after image is displayed in the lightbox, "next" and "previous" buttons will be displayed.

11. Slideshow

This chapter refers to the following section of the theme admin panel:

System > Configuration > Slideshow

Description of all the settings of the Slideshow available in the admin panel can be found later in this chapter: [11.7 Slideshow configuration](#).

11.1 Slideshow on home page

The slideshow can be added **to the home page** in two ways:

1. **Through the module admin panel** – in this field: *System > Configuration > Slideshow > General > Main Slideshow on the Home Page > Slides*.
2. **Through the shortcode** – the shortcode which adds the slideshow is described in the next chapter.

Both ways are described below.

11.1.1 Add home page slideshow through the module admin panel

In the *Slides* field in the admin panel under *System > Configuration > Slideshow > General > Main Slideshow on Home Page*, you need to enter an **identifier of a static block** which will contain your slides. You have two options:

- You can leave the identifier of the default static block `sample_slideshow_1` which contains a few sample slides.
- Or enter an identifier of your own static block with slides.

Whichever option you choose, go to *CMS > Static Blocks*, open your static block and edit its content to change the content of your slides.

For more information about creating content of the slides, refer to chapter [11.3 Create](#)

slides. You can also find there information about sample static blocks (with sample slides) which can be **imported** after theme installation.

11.1.2 Add home page slideshow through the shortcode

Second method to add a slideshow to the home page is to use the shortcode which is described with more details in chapter **11.2 Slideshow on any page**.

Go to *CMS > Pages*, open your home page and enter the shortcode in the main content field. For example, add it at the very top of the page content – this way your slideshow will be displayed just below the header on the home page. Magento will **automatically replace the shortcode with a slideshow** during page rendering.

11.2 Slideshow on any page

Slideshow can be displayed on any CMS page (and inside any static block). To add a slideshow to a page, use the following shortcode:

```
 {{block type="ultraslideshow/slideshow" template="infortis.ultraslideshow/slideshow.phtml"
    slides="sample_slideshow_1"}}
```

Enter the shortcode in the content field of your page, Magento will **automatically replace the shortcode with a slideshow** during page rendering.

The **slides** parameter must contain an identifier of a single static block which contains your slides. The slides from that static block will be displayed in the slideshow. You can also leave the default static block `sample_slideshow_1`, which contains a few sample slides for quick start.

To create the content of the slideshow (the slides), refer to chapter **11.3 Create slides** for more details. You can also find there information about how to **import sample static blocks** (with sample slides).

You can also import a sample CMS page which contains a few examples of slideshow shortcodes. Refer to chapter **11.5.2 Import page with examples of shortcodes** for

details about the import.

11.2.1 List of parameters

The shortcode which creates a slideshow can take a few parameters. Only the `slides` parameter is required. The other parameters are optional, they can be used to override the global settings of the Slideshow module. If you create multiple slideshows on a single page (by adding multiple shortcode), then in each shortcode you can have different values of parameters.

```
 {{block type="ultraslideshow/slideshow" template="infortis.ultraslideshow/slideshow.phtml"
slides="sample_slideshow_1" timeout="15000"}}
```

All available parameters are presented below.

Required parameters:

- **slides** – an identifier of a single **static block** which contains your slides, for example: `sample_slideshow_1`. To create slides, the content of that static block needs to have proper structure, refer to **11.3.1 Basic structure of the static block with slides** for more details.

Alternatively, you can also enter comma separated list of static block identifiers (but this option **is deprecated and will be removed in the future versions**), for example: `block_slide1, block_slide2, block_slide3`. In this case, the content of **each block** will be displayed **as a separate slide**.

Optional parameters:

- **timeout** – to animate the slideshow automatically, specify the time (in milliseconds) between slide transitions. Enter `0` to disable automatic transition. Note that 1000 milliseconds = 1 second.
- **slide_transition** – the effect to be used for transition between two slides. Available values:
 - `fade`
 - `backSlide`

- goDown
- fadeUp

Note:

Enter 0 to use simple horizontal slide animation: `slide_transition="0"`

- **pagination** – display or remove pagination. Allowed values:
 - 0 – remove pagination.
 - 1 – display pagination, visual style 1.
 - 2 – display pagination, visual style 2.
- **pagination_position** – select position of the pagination. Allowed values:
 - bottom-centered – bottom, centered, below the slides.
 - bottom-left – bottom, left, below the slides.
 - bottom-right – bottom, right, below the slides.
 - over-bottom-centered – bottom, centered, over the slides.
 - over-bottom-left – bottom, left, over the slides.
 - over-bottom-right – bottom, right, over the slides.
- **loop** – enter 1 if the slideshow should loop.
- **banner** – an identifier of a single **static block** which contains additional banners (or any other content) displayed at the side of the slideshow. For example: `sample_slideshow_additional_banners_1`.

To create additional side banners, the content of that static block needs to have proper structure, refer to **11.4 Create additional side banners** for more details.

Note:

Do not use this parameter if the slideshow will be displayed in a full width container.

- **banner_position** – position of the additional banners displayed at the side of the slideshow. Available values:
 - `right` – the additional banners will be displayed at the right side of the slideshow. This is the default value, if you want the additional banners to be displayed at the right side of the slideshow, you can omit this parameter.
 - `left` – the additional banners will be displayed at the left side of the slideshow.
- **gutter** – enter `0` to remove the space between the slides and the additional banners.

11.3 Create slides

Content of the slideshow (the slides) needs to be placed inside a **static block**. Follow these simple steps to create a block with slides:

1. Create a static block

So, to create the slides, first you need to create a static block. To do that, go to CMS > Static Blocks. Refer to [4.8.2 Static blocks](#) for more information about creating static blocks in Magento.

Important:

when you're creating a static block, make sure to set it as **Enabled** and assign it to a proper **store view**. Otherwise the static block will not be used in the slideshow even if the slideshow was configured correctly.

2. Or import static blocks

Alternatively, you can also import sample static blocks (with sample slides) which contains dozens of ready-to-use code examples. Refer to chapter **11.5.1 Import static blocks with sample slides** for more details about the import.

For example, the main slideshow from the home page of **demo 1** have the following identifier: `sample_slideshow_1`.

3. Add content

When the static block is ready, add some **content**.

The content of the static block with slides needs to have a **proper structure** – refer to chapter **11.3.1 Basic structure of the static block with slides** for more details.

4. Enter the identifier of that static block

Finally, to "tell" Magento where your slides are stored, the **identifier** of your static block needs to be entered in:

- the `slides` parameter, if you decided to create a slideshow through a shortcode (which is described with more details in chapter **11.2 Slideshow on any page**).
- or in the `Slides` field in admin panel under `System > Configuration > Slideshow > General > Main Slideshow on Home Page`, if you decided to create the main slideshow on the home page.

11.3.1 Basic structure of the static block with slides

The content of the static block with slides **needs to have the following structure**:

```
<div class="item">
    This is slide 1...
</div>

<div class="item">
    This is slide 2...
</div>

<div class="item">
```

```
This is slide 3...
</div>
```

As you can see, the content of the static block is **divided into items** (slides). Each item is a simple `<div>` element with the `item` class. Each item will be displayed as a separate slide in the slideshow. You can add as many items as you want.

Note:

make sure to close each item with the closing `</div>` tag. Otherwise the slideshow on frontend will be broken (e.g. two slides will be merged into one slide, the slideshow may be misplaced on the page, content of the page below the slideshow will be broken etc.).

To add custom content inside the first slide, replace the sample text `This is slide 1...` with any other content. For example, replace it with a simple image banner:

```
<div class="item">
    <div class="ban">
        
    </div>
</div>

<div class="item">
    This is slide 2...
</div>

<div class="item">
    This is slide 3...
</div>
```

Then, add another banner – replace the sample text `This is slide 2...` with an image banner:

```
<div class="item">
    <div class="ban">
        
    </div>
</div>

<div class="item">
    <div class="ban">
        
    </div>
</div>
```

```

        </div>
    </div>

<div class="item">
    This is slide 3...
</div>
```

And so on. You can create as many slides as you want (to read more about creating content of slides, refer to the next chapter: [11.3.2 Content of a slide](#)).

You need to use this structure if you keep your slides in a **single static block**, i.e. if you entered an identifier of a **single static block** in the `slides` parameter of the slideshow shortcode (or in the `Slides` field in admin panel under *System > Configuration > Slideshow > General > Main Slideshow on Home Page*).

But if you decided to keep **each of your slides in a separate static block**, i.e. if you entered a comma separated **list of static block identifiers** in the `slides` parameter of the slideshow shortcode (or in the `Slides` field in admin panel under *System > Configuration > Slideshow > General > Main Slideshow on Home Page*), then **you don't use this structure**. However, the option with comma separated list of static block identifiers is deprecated – since version 1.18.0 it's recommended to keep all slides in a single static block (for clarity and for better performance).

11.3.2 Content of a slide

Inside each slide you can use any HTML: text, images, icons etc. In most cases slides contain image banners with additional captions, so to create slides you can use the built-in system of banners which offers lots of useful features. Full description of banners with many examples can be found in chapter [6.5.5 Banners](#).

For example, to create a clickable image banner inside a slide, use the following markup:

```

<a href="{{store url="about-us"}}>
    <div class="ban">
        
    </div>
</a>
```

Important:

a slideshow is always only as big as the container in which it was placed. Even if you put very large images inside your slides, the **visual size** of your slideshow will be always limited by the size of the containing block.

11.3.3 Animated content of a slide

Inside each slide you can use any HTML: text, images, icons etc. In most cases slides contain image banners with additional captions, so to create slides you can use the built-in system of banners which offers lots of useful features. Full description of banners with many examples can be found in chapter **6.5.5 Banners**.

When a banner is used in a slide, it can be easily extended with eye-catching **animations**. In this chapter you can find a few simple examples.

Note:

this chapter describes the animations which can be added **to the content of a slide** (e.g. to the text inside a slide). Do not confuse this with the animations which are used for transitions between slides – the slide transition animation can be configured globally in the *Transition Effect* field under *Stores > Configuration > Ultimo > Slideshow > General*.

Here's an example of a banner with caption (a heading and two lines of text). This is our basic code which will be extend with animations (it doesn't have any animations yet):

```
<div class="ban">
    
    <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
        <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
        <p class="text" data-animate-in="fadeInRight delay-1">This is a line of text</p>
        <p class="text" data-animate-in="fadeInRight delay-2">Second line of text</p>
    </div>
</div>
```

To add an animation effect, add the `data-animate-in` attribute to selected line of text and

specify the class name of the animation, for example: `fadeInRight`. In the example below, the animation was added to the heading element:

```
<div class="ban">
    
    <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
        <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
        <p class="text">This is a line of text</p>
        <p class="text">Second line of text</p>
    </div>
</div>
```

The `fadeInRight` animation class will make the text fade in and move horizontally from right to left when the slide is first shown on the screen. The list of all available animation classes can be found in chapter [6.5.6.B List of animations](#).

Note:

note that we add the animation name to the `data-animate-in` attribute of the selected element, **not to the `class` attribute**. The slideshow script will automatically add the animation name to the `class` attribute when the slide is shown on the screen (so you don't need to worry about how and where the animation classes will be actually used).

We can now add the same animation effect to the rest of the text elements:

```
<div class="ban">
    
    <div class="cap cap-top-left cap-no-bg cap-text-bg cap-text-bg-dark-1">
        <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
        <p class="text" data-animate-in="fadeInRight">This is a line of text</p>
        <p class="text" data-animate-in="fadeInRight">Second line of text</p>
    </div>
</div>
```

To make the animation more eye-catching we can use the delay classes (such as `delay-1` and `delay-2`). These classes delay the animation start. The number at the end of each class name indicates the value (in seconds) by which the animation will be delayed.

Let's try these useful classes. Add `delay-1` to the first line of text (this will add 1 second of delay) and `delay-2` to the second line of text (this will add 2 seconds of delay):

```
<div class="ban">
    
    <div class="cap cap-push-up-10 cap-push-right-10 cap-no-bg cap-text-bg cap-text-bg-dark-1">
        <h3 class="text" data-animate-in="fadeInRight">Caption Example</h3>
        <p class="text" data-animate-in="fadeInRight delay-1">This is a line of text</p>
        <p class="text" data-animate-in="fadeInRight delay-2">Second line of text</p>
    </div>
</div>
```

More utility classes for animations can be found in chapter [6.5.6.C Additional options](#).

11.3.4 Size of slides and banners

The slideshow is responsive so the **width of the slideshow will expand to fit the width of the content area of the page** (however, the height of the slideshow is always the same as the height of the images inside the slides).

Note:

the width of the content area of the page depends on the value of the *Maximum Breakpoint* field (under *Stores > Configuration > Theme Layout*). More details can be found in chapter [9. Theme Layout](#). So when you create images for your slides, you need to adjust the image size to the maximum width of the content area.

Inside your slides you can use images of any size. If an image is larger than the width of the slide, it will be automatically scaled down to fit the available space.

But to achieve the best visual effect it's recommended to use image sizes presented in the table below. It shows the width of slides (in pixels) for each available value of the *Maximum Breakpoint* field.

For example, for the default value of the *Maximum Breakpoint* (which is `1200 px`) and the default value of the *Side Padding* (which is `15 px` and can be found under *System > Configuration > Theme Design > Page > Inner Container*), the recommended width of an image inside a slide is `1140 px`:

| Maximum Breakpoint | Slide width |
|--------------------|-------------|
| 992 | 940 |
| 1200 | 1140 |
| 1440 | 1350 |
| 1680 | 1470 |
| 1920 | 1710 |

Height of the slideshow is not limited so you can use images of any height. However, to get consistent look, it's recommended to use images of the same height in all slides.

11.3.4.A Slideshow with additional side banners

If the slideshow has additional side banners, it's recommended to use image sizes presented in the following table:

| Maximum Breakpoint | Slide size | Additional side banners | |
|--------------------|------------------------------|-------------------------|-------------------------|
| | | Banner size | Banner size (no gutter) |
| 992 | 690 x 333 | 220 x 101 | 235 x 111 |
| 1200 | 855 x 390 | 270 x 120 | 285 x 130 |
| 1440 | 997 x 390 | 322 x 120 | 337 x 130 |
| 1680 | 1102 x 420 | 352 x 130 | 367 x 140 |
| 1920 | side banners not recommended | | |

In this table we assume that you use three equal banners which are displayed one below the other.

If the main slides and side banners are **not separated with a gutter**, for the side banners use the sizes from the column "*Banner size (no gutter)*". But if the main slides and side banners **are separated with a gutter**, the side banners needs to be a bit smaller (because they also have a bottom gutter) so use the sizes from the column "*Banner size*".

11.4 Create additional side banners

For every slideshow you can create additional small banners displayed at the side of the slideshow. In this example we will create three equal banners which are displayed one below the other.

The content of the static block with the additional side banners **needs to have the following structure:**

```
<div class="row row-bottom-gutter-half">  
    <div class="col-md-12 small-banner">  
        This is unit 1...  
    </div>  
  
    <div class="col-md-12 small-banner">  
        This is unit 2...  
    </div>  
  
    <div class="col-md-12 small-banner">  
        This is unit 3...  
    </div>  
</div>
```

To build the basic structure, we use the grid system. Each unit is a simple `<div>` element with the `small-banner` class and with the grid class (in this case we use the `col-md-12` class).

Note:

make sure to close each element with the closing `</div>` tag. Otherwise the slideshow on frontend will be broken (e.g. banners will be misplaced, content of the page below the slideshow will be broken etc.).

To add small bottom gutter between units, add the `row-bottom-gutter-half` class to the row element. Remove this class if you don't want to have any space between units.

To add a banner inside the first unit, replace the sample text `This is unit 1...` with the code of a simple image banner. For example:

```
<div class="row row-bottom-gutter-half">  
    <div class="col-md-12 small-banner">  
        <div class="ban ban-effect-1">  
              
        </div>  
    </div>  
  
    <div class="col-md-12 small-banner">  
        This is unit 2...  
    </div>
```

```
</div>

<div class="col-md-12 small-banner">
    This is unit 3...
</div>

</div>
```

Then, add another banner:

```
<div class="row row-bottom-gutter-half">

    <div class="col-md-12 small-banner">
        <div class="ban ban-effect-1">
            
        </div>
    </div>

    <div class="col-md-12 small-banner">
        <div class="ban ban-effect-1">
            
        </div>
    </div>

    <div class="col-md-12 small-banner">
        This is unit 3...
    </div>

</div>
```

Finally, add the third banner:

```
<div class="row row-bottom-gutter-half">

    <div class="col-md-12 small-banner">
        <div class="ban ban-effect-1">
            
        </div>
    </div>

    <div class="col-md-12 small-banner">
        <div class="ban ban-effect-1">
            
        </div>
    </div>

    <div class="col-md-12 small-banner">
        <div class="ban ban-effect-1">
            
        </div>
    </div>

</div>
```

```
</div>  
</div>
```

11.5 Import sample slides and shortcodes

11.5.1 Import static blocks with sample slides

For quick start, you can import sample static blocks which contain ready-to-use examples of slides. Refer to chapter [7.12 Data Import](#) for details about the import procedure.

After import, you can find all imported static blocks under [CMS > Static Blocks](#). Identifiers of the static blocks with sample slides start with the `sample_slideshow_` prefix. For example, the main slideshow from the home page of **demo 1** have the following identifier: `sample_slideshow_1`.

You can modify the imported static blocks to create your own slides for slideshows.

11.5.2 Import page with examples of shortcodes

You can import a sample CMS page which contains a few examples of slideshow shortcodes. Refer to chapter [7.12 Data Import](#) for more details about the import procedure.

The identifier of the page is: `magento-banner-slideshow`. After import, the page can be found under [CMS > Pages](#).

Page contains examples of shortcodes which create slideshows of different sizes. More information about size of slides and banners can be found in [11.3.4 Size of slides and banners](#).

Important:

To use these shortcodes first you need to import sample static blocks with sldies.

11.6 Full width slideshow on home page

To add a full width slideshow to the home page, follow the instructions in chapter [11.1.1 Add home page slideshow through the module admin panel](#).

When the slideshow is ready, to make it full width, follow these steps:

1. Go to *System > Configuration > Slideshow > Main Slideshow on the Home Page*.
2. Enable the *Position: Above Content Area (Full Width)* option.
3. **Disable** the *Position: Above Content Area (Boxed)* option. It has to be disabled because the slideshow can't be added to both positions at the same time.
4. Finally, flush the cache.

The main slideshow on the home page will be displayed at the top of the main content area, just below the header. The slideshow will take full available width.

List and description of all the options available in the admin panel can be found in chapter [11.7 Slideshow configuration](#).

11.7 Slideshow configuration

Below you can find a list of all slideshow settings available in the admin panel in *System > Configuration > Ultimo > Slideshow*.

11.7.1 General

Main Slideshow on the Home Page

In this tab you can quickly configure and enable the main slideshow on the home page. Settings from this section will be applied **only to the main slideshow on the home page**. To display slideshows in other places and on other pages, use the shortcode described earlier in this chapter.

- **Slides** – enter an identifier of a static block which contains your slides. The slides from that static block will be displayed in the main slideshow on the home page. You can use the default static block `sample_slideshow_1` which contains a few sample slides for quick start.

Important:

Leave this field empty to remove the main slideshow from the home page.

Alternatively you can also enter a comma separated list of static block identifiers. Content of each block will be displayed as a separate slide. Note that **this option is deprecated and will be removed in the future versions**, we recommend to enter only a single static block which contains all slides.

- **Position: Above Content Area (Boxed)** – if enabled, the main slideshow will be displayed at the top of the main content area of the home page (just below the header). Slideshow will be boxed – it will have the same width as the content area and sidebars.

Note:

Do not enable slideshow in both positions at the same time.

- **Position: Above Content Area (Full Width)** – if enabled, the main slideshow will be displayed at the top of the main content area of the home page (just below the header). **Slideshow will take the full width of the viewport.**

Note:

Do not enable slideshow in both positions at the same time.

- **Hide Slides On Mobile Devices** – if set to **Yes**, slideshow will be hidden on mobile devices (if browser viewport width is less than 768 pixels). Important: the additional banners displayed at the side of the slideshow can be hidden separately, see options described later in this chapter.
- **Top Margin** – additional top margin for the slideshow. You can set margin to add extra space above the slideshow. Enter negative value to pull the slideshow up.
- **Bottom Margin** – additional bottom margin for the slideshow. You can set margin to add extra space below the slideshow.

General Settings

In this tab you can configure basic settings of a slideshow, such as speed and timeout. These settings will be applied to all slideshows on frontend.

- **Transition Effect** – the effect which will be used for transition between two slides.

Note:

Leave empty to use simple horizontal slide animation.

- **Timeout** – to animate the slideshow automatically, specify the time (in milliseconds) between transitions. Enter **0** to disable automatic transition. Note that 1000 milliseconds is equal to 1 second.
- **Scrolling Speed** – duration (in milliseconds) of horizontal scrolling animation. Defines how fast the current slide is replaced with the next slide. Applied to horizontal scrolling with arrows and touch swipe.
- **Automatic Scrolling Speed** – duration (in milliseconds) of horizontal scrolling

animation. Applied to automatic horizontal scrolling (if *Timeout* was specified) and to horizontal scrolling with pagination.

- **Smooth Height** – enable, if slides do not have the same height. It will allow height of the slideshow to animate smoothly.
- **Pause on Hover** – if enabled, when the mouse is over the slideshow the automatic transition functionality will pause.
- **Loop** – if enabled, slideshow will automatically loop.

11.7.2 Additional Banners

In this tab you can configure additional banners displayed at the side of the main slideshow on the home page. These settings are applied **only to the main slideshow on the home page**.

- **Banners** – enter an identifier of a single static block which contains additional side banners for the slideshow. These banners will be displayed at the side of the slideshow.

Note that these banners will be applied **only to the main slideshow on the home page**. To add side banners to other slideshows, use proper parameter in the shortcode of each slideshow (the shortcode is described earlier in this chapter).

You can use the default static block

`sample_slideshow_additional_banners_1`, which contains sample banners for quick start.

Important:

Leave this field **empty** if you don't want to add side banners to the main slideshow on the home page.

- **Position of Banners** – select position of the banners (left or right).
- **Show Gutter Between Slides and Banners** – if set to `Yes`, there will be space between the slides and the additional banners.
- **Hide Banners On Mobile Devices** – if set to `Yes`, banners will be hidden on mobile devices (if browser viewport width is less than 768 pixels).

11.7.3 Navigation

In this tab you can configure slideshow navigation elements. These settings will be applied to all slideshows on frontend.

- **Navigation Buttons** – enable/disable "next" and "previous" buttons.
- **Pagination** – enable/disable and select style of pagination.
- **Pagination Position** – select position of the pagination.

12. Brands

This chapter refers to the following section of the theme admin panel:
System > Configuration > Brands

List of all settings available in the admin panel can be found later in this chapter:
12.2 Brands configuration.

12.1 Basic information

Below you can find some basic information about brand configuration.

Important:

after you finish configuration of all settings, remember to reindex the data (**4.2 Index management**) and flush the cache (**4.1 Caching system**).

12.1.1 Brand attribute

First of all, to show brands, you will need an attribute which will store the information about brands. By default there is an attribute `manufacturer` in Magento which you can use for this (but you can of course create your own attribute, if needed).

If you decide to create your own brand attribute, the most important thing is to select proper type. There are many types of attributes in Magento, but for brands you need to select `drop-down` in the **Catalog Input Type for Store Owner** field. This type of attribute will let you define a list of values (list of brands) available for products. For more information about attribute creation and management refer to **4.7 Attributes**.

12.1.2 Brand images

When your attribute is ready and was added to products, using your FTP client upload your brand logo images to the following directory: media/wysiwyg/infortis/brands/. Optimal size of the image is 210 x 50 pixels. All images should have the same file extension (the extension can be specified in brand settings described later in this chapter).

12.1.3 Brand names and image file names

All images should be named as per the entries in the brand attribute, but the names of the files has to be simplified a little bit.

| Admin | English | French | German | Position | Is Default | Action |
|-----------|---------|--------|--------|----------|-----------------------|---------------------------------------|
| Acer | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| AMD | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Anashria | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Apple | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Argus | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Asics | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Canon | | | | 0 | <input type="radio"/> | <input type="button" value="Delete"/> |
| Crucial | | | | 10 | <input type="radio"/> | <input type="button" value="Delete"/> |
| New Brand | | | | 20 | <input type="radio"/> | <input type="button" value="Delete"/> |

Below you can find the list of rules which you need to follow in order to convert a brand name to a correct file name. The module will follow the same rules to generate the path of the image file for each brand. If files are not named correctly, then the module will not be able to find and display brand images in your store.

The same rules will be used to generate links to custom pages if you configure brand logos to be links to custom pages.

Note:

in exactly the same way Magento simplifies category names and product names to automatically create **URL Key** for category pages and product pages. Full symbol conversion table can be found in Magento file: app/code/core/Mage/Catalog/Helper/Product/Url.php

Brand name conversion rules:

1. Replace all uppercase letters with lowercase letters, e.g.:

for brand BlackBerry create an image named like this → blackberry.png

2. Define the word separator (for simplicity we will call it **the separator**) in the **Image URL Key Separator** field in *System > Configuration > Brands > General* section of admin panel. The separator will be used to replace all whitespace characters and other non-letter and non-digit characters in file names.

It is recommended to use the default separator: a single hyphen (-). More info about separator can be found later in this chapter: **12.2 Brands configuration**.

3. Replace single space with the separator, e.g.:

Louis Vuitton → louis-vuitton.png

4. Replace single special characters (non-letter and non-digit characters such as apostrophe, dot, ampersand, percent sign, exclamation mark etc.) with the separator, e.g.:

McDonald's → mcdonald-s.png

AT&T → at-t.png

Amazon.Com → amazon-com.png

Coca-Cola → coca-cola.png

5. Replace series of whitespace and special characters with a single separator, e.g.:

Marks & Spencer → marks-spencer.png

J.P. Morgan → j-p-morgan.png

6. Replace language specific symbols and characters (e.g. letters with diacritical marks) with letters from latin alphabet, e.g.:

Huáwéi → huawei.png

Citroën → citroen.png

Crédit Agricole → credit-agricole.png

12.2 Brands configuration

Below you can find a list of settings available in the admin panel *System > Configuration > Brands*.

12.2.1 General

- **Brand Attribute Code** – specify the **Attribute Code** of the attribute which stores the information about brands in your Magento (you can use Magento's default attribute `manufacturer` or create your own attribute).

If such attribute doesn't exist in your store, you will need to create it and add it to the Attribute Sets which you use for your products.

To find the **Attribute Code** of existing attribute go to *Catalog > Attributes > Manage Attributes*, click on the attribute and search the **Attribute Code** field:

- **Brand Image File Extension** – specify the file extension (jpg, png or gif) of your brand logo images.
- **Show Brand Image** – if set to `No`, brand image will be replaced with brand name (simple text). This option is useful when you don't want to show brand images, only the brand names.
- **Show Text If No Brand Image** –
 - if set to `Yes`: if brand image doesn't exist, it will be replaced with brand name (simple text).

- if set to **No**: if brand image doesn't exist, it will be not displayed and there will be no other information about brand.

Link

- **Brand Logo is a Link to...** – with this option, you can decide whether brand logo is a link (e.g. to search results page).
 - **No Link** – select this option if you don't want brand logo to be a link.
 - **Quick Search Results** – brand logo is a link to quick search results page. For example: logo of brand *Apple* on product page and in brand sliders will be a link to the following page:
`www.example.com/catalogsearch/result/?q=apple`which displays a list of products returned by Magento quick search for term "Apple".
 - **Advanced Search Results** – brand logo is a link to advanced search results page which displays a list of products returned by Magento advanced search for brand attribute value "Apple".
 - **Custom Page** – brand logo can also be a link to custom pages (e.g. category view pages, CMS pages, pages created by 3rd-party extensions etc.). In that case, each brand logo on product page and in brand sliders will be a link to a custom page.

Note:

this option doesn't create any custom pages for brands. It only makes it possible to link your brand logos to pages such as: category view pages, CMS pages or other pages created by 3rd-party extensions.

After selecting this option, more settings will appear below this field.

- **Base Path of Brand Pages** – path specified in this field will be used to build a link to a custom brand page (for each brand on product page and in brand sliders). For example, brand logo can be a link to a category which contains products from that brand. It can also be a link to any CMS page you want. To use this option select **Custom Page** in **Brand Logo is a Link to...** field.

Path will be appended to the store base URL before the brand name.

For example: `www.example.com/custom/base/path/apple`

| | | |
|-------------------------------|---------------------------------|---------------------|
| <code>www.example.com/</code> | <code>custom/base/path</code> | <code>/apple</code> |
| <i>Store base URL</i> | <i>Base Path of Brand Pages</i> | <i>Brand name</i> |

Here you can see some examples.

- **Example 1:**

You sell smartphones and you want *Apple* logo to be a link to a category with other Apple products.

- You created top-level categories for all your brands. If the category page URL is `www.example.com/apple`, then you should leave the **Base Path of Brand Pages** field empty.
- You created sub-categories for all your brands. They are sub-categories of the main category named “All Brands” and the URLs of category pages are the following:
 - `www.example.com/all-brands/apple`,
 - `www.example.com/all-brands/samsung` etc.
 In that case you should enter `all-brands` in the **Base Path of Brand Pages** field. Thanks to that the module will be able to build correct links to category pages for all your brands.

- **Example 2:**

You sell smartphones and you want *Apple* logo to be a link to a custom CMS page with some additional info about the brand. In Magento custom pages can be easily created in `CMS > Pages`. For each page you need to specify value of **URL Key** field which is a unique identifier of a page. E.g., custom page with URL Key `apple` will have the following URL: `www.example.com/apple`

- You created custom pages for all your brands. If the URLs of your custom pages are the following:
 - `www.example.com/apple`,
 - `www.example.com/samsung` etc.,
 then you should leave the **Base Path of Brand Pages** field empty.
- URL Key of a custom page can be more complex than just a single word. All your brand pages can have some common part, e.g., `my-brands/apple`, `my-brands/samsung` etc. You created custom pages for all your brands.

The URLs of your custom pages are the following:

- www.example.com/my-brands/apple,
- www.example.com/my-brands/samsung etc.

In that case you should enter `my-brands` in the **Base Path of Brand Pages** field. Thanks to that the module will be able to build correct links to custom pages for all your brands.

- **Append Category URL Suffix** – if **Category URL Suffix** exists (in *System > Configuration > Catalog*), it can be appended at the end of the brand link. For example: if `.html` was set as the **Category URL Suffix**, logo of brand *Apple* on product page and in brand sliders will be a link to the following page:

`www.example.com/custom/base/path/apple.html`

SEO Settings

- **URL Key Separator** – specify character which will be used as word separator in URLs of brand pages. Hyphen (`-`) is the recommended character. Other separators, such as underscores (`_`) or plus-signs (`+`) should be avoided. More information: <https://www.youtube.com/watch?v=AQcSFsQyct8>

Module will strip all whitespace characters and other non-letter and non-digit characters and replace them with the separator. For example:

if brand name is `Marks & Spencer` and separator is hyphen (`-`), the brand name in URL will be the following: `marks-spencer`

More information about how brand names are converted (to generate name of the brand image file or brand page URL) can be found in **12.1.3 Brand names and image file names**.

- **Image URL Key Separator** – specify character which will be used as word separator in the names of brand image files. For example:

if brand name is `Louis Vuitton`, separator is hyphen (`-`) and file extension is `png`, the name of the brand image file will be the following: `louis-vuitton.png`

Important:

the **Image URL Key Separator** field was added for backward compatibility. In next versions this field will be removed and the character specified in the **URL Key Separator** field will also be used in the names of brand image files.

To prepare your store for this, we recommend to use hyphen (`-`) as the

separator for images. In previous versions underscore () was the default separator for images. So what you need to do is to enter hyphen in the Image URL Key Separator field and edit names of all your brand images: replace underscores with hyphens.

For example: change file name `louis_vuitton.png` to `louis-vuitton.png`

12.2.2 Brand on Product Page

- **Show Brand on Product Page** – if enabled, will show a brand logo on the product page (if product has an attribute which specifies the brand).

12.2.3 Brand List

Select Brands

In this tab you can select list of brands which will be displayed in brand sliders. If you have multiple sliders (e.g., on different pages of the store), then each slider will display the same list of brands.

Note:

you can also override the list of selected brands by using `brands` parameter in the slider block code – parameter is described later in this chapter. With `brands` parameter it is possible to display different set of brands in each slider.

- **All Brands** – if set to Yes, all existing brands will be displayed. If set to No, you will be able to select brands in the field below.
- **Selected Brands** – select brands which you want to display.

Filter Brands

- **Brands Assigned To Products** – if set to Yes, display only those brands, which are currently assigned to products.
- **Brands Assigned To Products In Stock** – if set to Yes, display only those brands, which are currently assigned to products which are in stock.

12.2.4 Brand Slider

Brand slider can be displayed on any CMS page and in any static block. For more details see the block code which is described later in this chapter.

Below you can find a list of global settings for all brand sliders, some of the settings can be overridden per slider by using parameters which are described later in this chapter.

- **Show All Brands** – if set to **Yes**, slider will display all existing brands. Otherwise slider will display only those brands, which are currently assigned to products.
- **Timeout** – Time (in milliseconds) between slide transitions. Enter **0** to disable automatic transition. Note that 1000 milliseconds is equal to 1 second.
- **Speed** – duration (in milliseconds) of scrolling animation. Applied to scrolling with arrows (and touch swipe).
- **Automatic Scrolling Speed** – duration (in milliseconds) of scrolling animation. Applied to automatic scrolling (if **Timeout** was specified) and to scrolling with pagination.
- **Pause on Hover** – if enabled, when the slider is mouseovered then the automatic scrolling will pause.
- **Loop** – if enabled, scrolling will loop.
- **Lazy Loading** – delays loading of images. If enabled, images outside of viewport will not be loaded before user scrolls to them.

Brand slider block code and parameters

To display brand slider, enter the following code in the content field of any page or static block, Magento will automatically replace the code with a slider during page rendering:

```
 {{block type="brands/brands" template="infortis/brands/brand_slider.phtml"
breakpoints="[0, 1], [320, 2], [480, 2], [768, 3], [992, 4], [1200, 5]" block_name="Our
Brands"}}
```

Here is the list of available parameters. Some of them can override global settings:

Required parameters:

- **block_name** – the name of the block displayed as a heading.

Optional parameters:

- **brands** – with this parameter you can specify the list of brands (separated by comma) which will be displayed in the slider. You can have multiple sliders and each can display completely different set of brands – just provide different list of brands in each slider code using this parameter, for example:

```
brands="Apple, Samsung, Louis Vuitton, Coca-Cola"
```

This list will override the list of brands which was selected directly in the admin panel in *System > Configuration > Brands > Brand List* section. But if this parameter is not specified, slider will display brands selected in the admin panel.

- **breakpoints** – with this parameter you can specify the number of items in a row visible with a particular browser viewport width. Use the following format (this is the default value, it will be used if this parameter is not specified in the block code):

```
breakpoints="[0, 1], [320, 2], [480, 2], [768, 3], [992, 4], [1200, 5]"
```

Value of this parameter consists of pairs of numbers in brackets separated by comma. In each pair of numbers `[A, B]` number `A` indicates the web browser viewport width, number `B` indicates the number of items visible if browser viewport width is greater than the width specified in `A`.

In other words: display `B` items if viewport width is above `A` pixels. Below you can find some examples:

- **Example 1:**

```
breakpoints="[0, 2], [480, 3], [992, 5]"
```

which can be translated as: display 2 items if viewport width is between 0px and 480px, display 3 items if viewport width is above 480px, display 5 items if viewport width is above 992px.

- **Example 2:**

```
breakpoints="[0, 4]"
```

which can be translated as: always display 4 items.

- **Example 3:**

```
breakpoints="[0, 3], [992, 6]"
```

which can be translated as: display 3 items on narrow screens (if viewport width is below 992px), and display 6 items on wider screens (above 992px).

- **is_responsive** – enter `0` if the slider **should not be responsive**. If this parameter is not specified, it is always assumed that the slider is responsive.

Important:

If the slider **is not** responsive, the number of visible items should be specified with the additional **show_items** parameter (see below).

- **show_items** – the number of products in a row in the non-responsive slider. Use this parameter only if responsive behavior was disabled with parameter **is_responsive** (see above)
- **timeout** – to animate the slider automatically, specify the time (in milliseconds) between transitions. Note that 1000 milliseconds = 1 second.
- **move** – number of items that should move on animation. Allowed values:
 - enter `1` to move one item.
 - enter `0` to move all visible items. This is also the default value which is used if the parameter is not specified.
- **loop** – enter `1` if the slider should loop.
- **pagination** – enter `1` to display pagination.

13. Menu

This chapter refers to the following sections of the admin panel:

- General menu settings:

System > Configuration > Menu

Description of general settings can be found in this chapter:

13.1 Menu configuration

- Menu settings which can be applied individually for each category:

Catalog > Category Management, in the *Menu* tab.

Description of individual category settings can be found in this chapter:

13.2 Drop-down boxes

More information about category management in Magento can be found in chapter [4.5 Categories](#).

Note:

In Magento you can limit number of category levels displayed in the menu. Refer to [4.5.4 Depth of the menu](#) for more details.

13.1 Menu configuration

Below you can find a list of the menu settings available in the admin panel *System > Configuration > Menu*.

13.1.1 Top Menu

- **Top Menu** – enable/disable the horizontal drop-down menu in the header section of the page.
- **Mode** – select menu mode.
 - **Drop-down** – standard drop-down menu.
 - **Mobile** – collapsible accordion dedicated for mobile websites.
 - **Drop-down/Mobile** – combination of both modes, dedicated for responsive websites. You can specify the width of the browser viewport below which the drop-down menu turns into the mobile menu (see the **Threshold** field under the **Mobile Menu** tab).
- **Display Categories** – display categories in the top menu.

Vertical Layout

- **Vertical Drop-down Menu** – enable vertical layout of the menu.

In this layout, categories are stacked vertically inside a drop-down box which can be shown/hidden when user clicks the button inside the menu bar. Subcategories are displayed in drop-downs at the right side of the main drop-down.

Home Link

- **Home Link: Text** – if enabled, will display menu item with text "Home" as a link to the home page. This link will be displayed with the same colors as other menu items.
- **Home Link: Icon** – if enabled, will display menu item with icon as a link to the

home page. This link will be displayed with the same colors as other menu items.

- **Single Icon as Home Link** – enable to show single icon as a link to the home page. Icon will be displayed without any additional background color. If it's enabled, options *Home Link: Text* and *Home Link: Icon* will be ignored.

Additional Settings

- **Align Drop-down With...** – the drop-down box (of top-level category) will be aligned with the right edge of the selected container, if the right edge of the drop-down box sticks out further to the right than the selected container.
 - **Viewport** – drop-down box will be aligned with the user's visible area of the page.
 - **Menu bar** – drop-down box will be aligned with the right edge of the menu bar.
 - **Primary header, inner container** – drop-down box will be aligned with the inner container of the primary header section.
- **Full-width Drop-down As Wide As...** – full-width drop-downs can be as wide as the selected container.

If a drop-down box of a top-level category was set as full-width (i.e in category settings the *Submenu Type* field was set to *Mega drop-down* and the *Drop-down Width* field was left empty), it will be as wide as the selected container.

Note:

width of a drop-down box can be configured individually for each category in category settings.

- **Custom Static Block - Align Right** – align custom static block to the right side of the menu bar.

To add custom static block to the menu you need to create a static block with ID `block_header_nav_dropdown` under *CMS > Static Blocks*. Content of that static block will be displayed as a drop-down box.

Note:

there is another method to add custom drop-down boxes to the menu, for more details refer to **13.4 Custom drop-down boxes in the menu**.

13.1.2 Mobile Menu

- **Threshold** – width of the browser viewport (in pixels), below which the drop-down menu turns into a mobile menu (an accordion) dedicated for mobile devices. The threshold is needed only if the top menu is in the default **Drop-down/Mobile** mode.

Important:

The default value of the threshold:

- Horizontal menu:

for horizontal menu (when Vertical Drop-down Menu is not enabled) it's recommended to leave the default value of the threshold: 960. Values below 960 are not recommended - horizontal drop-down menus usually don't look good on narrow screens.

- Vertical menu:

if the Vertical Drop-down Menu is enabled, it's recommended to set the threshold to 770.

- **Show Category Blocks From X Levels of Categories** – specify number of levels of categories (counting from the top level) for which category blocks will be displayed in mobile menu.

E.g. enter **1** to display blocks of 1st-level categories, enter **2** to display blocks of 1st-level and 2nd-level categories, enter **3** to display blocks of 1st-level, 2nd-level and 3rd-level categories, and so on.

The allowed range is from **0** to **5**.

Important:

enter **0** to hide blocks of all categories on all levels in the mobile menu.

- **Hide Category Blocks Below 320px** – category blocks will be hidden in the mobile menu on small mobile devices (if width of the browser viewport is below 320px).

13.1.3 Sidebar Menu

In this section you can enable and configure the sidebar menu: vertical list of categories which can be displayed at the top of the sidebars on selected pages.

Note:

you can also display the sidebar menu inside any static block and on any CMS page by using the sidebar menu shortcode. The shortcode is described later in this chapter.

Enable Sidebar Menu

Here you can enable/disable sidebar menu in different locations of the store.

You can choose in which sidebar the menu will be displayed. Depending on the layout (one, two or three columns), each page might have one or two sidebars (or no sidebar at all). Layout can be configured individually for each page (e.g. in category's settings, in CMS page's settings etc.). To change the layout of a page, change the value of its *Page Layout* field.

Here is the list of options to enable the sidebar menu:

- **Category View (Left Sidebar)** – enable/disable menu in category view at the top of the left sidebar. It will be displayed above Magento's Layered Navigation – for more details about Layered Navigation refer to [4.5.5 Layered Navigation \("Shop By" filters\)](#).
- **Category View (Right Sidebar)** – enable/disable menu in category view at the top of the right sidebar.
- **Home Page (Left Sidebar)** – enable/disable menu on home page at the top of the left sidebar.
- **Home Page (Right Sidebar)** – enable/disable menu on home page at the top of the right sidebar.
- **CMS Pages (Left Sidebar)** – enable/disable menu on CMS pages at the top of the

left sidebar.

- **CMS Pages (Right Sidebar)** – enable/disable menu on CMS pages at the top of the right sidebar.
- **Product Page (Left Sidebar)** – enable/disable menu on product pages at the top of the left sidebar. Note that by default a product page have no sidebars, so you need to change the layout of a product page to display the sidebar menu. You can change the layout in product's settings.
- **Product Page (Right Sidebar)** – enable/disable menu on product pages at the top of the right sidebar. Note that by default a product page have no sidebars, so you need to change the layout of a product page to display the sidebar menu. You can change the layout in product's settings.

Sidebar Mega Menu

- **Sidebar Mega Menu** – enable sidebar mega menu. Sidebar mega menu can display drop-downs in a similar way to the top menu. Categories are stacked vertically and sub-categories are displayed in drop-downs at the right side of the menu.
If this option is not enabled, the sidebar menu will be displayed as a simple accordion. Additional settings for accordion can be found below.
- **Use Sidebar Mega Menu As Mobile Menu** – when the sidebar menu exists on the page, it will be used as mobile menu (when mobile menu is displayed).

Block Settings

- **Block Title** – the name of the block displayed at the top of the block.

Additionally you can use variable `[current_category]` inside the title to create dynamic name of the block. Variable will be replaced with the name of the current category. On pages other than category view or product view this variable will be empty.

Accordion Settings

- **Parent Category** – the menu will only contain children (subcategories) of the selected parent category. Available options:
 - **Root** – show all top-level categories in the store.

- **Parent of current category** – show the current category and its siblings (other categories from the same level as the current category).
- **Parent of current category (no siblings)** – show the current category. Other categories from the same level as the current category will be ignored.
- **Current category** – show subcategories of the current category.
- **Maximum Depth** – this parameter controls how many levels in the hierarchy of categories (starting from the parent category) are to be included in the menu. Menu with a Maximum Depth of **1** will just be an un-nested list of categories from a single level (with none of the subcategories displayed). The default value is **0**, it will display all subcategories.
- **Fallback To Root Category** – if current category doesn't exist, go back to root category. This option is useful if sidebar menu should be displayed on pages with no current category (on pages other than category view). If set to "Yes" and if current category doesn't exist, the menu will behave like **Parent Category** field was set to "Root" (it will display top-level categories). If set to "No", menu will not be displayed at all if there's no current category.

For example, if menu is displayed on the home page (**Parent Category** field is set to "Current category") and **Fallback To Root Category** field is set to "Yes", then the menu will display top-level categories.

If in the same example **Fallback To Root Category** field is set to "No", then the menu will not show any categories and will not be displayed at all.

- **Fallback Block Title** – this block title will be used instead of the standard block title, if **Fallback To Root Category** field is set to "Yes" and if current category doesn't exist.
- **Show Number of Products** – show number of products next to each category.

Additional Settings

- **Hide Sidebar Menu on Mobile Devices** – list of categories in the sidebar can be hidden on mobile devices (if browser viewport is narrower than 768 pixels). All categories are displayed in the top main menu by default so with this option you can save some space on mobile devices.
- **Hide Magento's Category Filter** – if you enabled sidebar menu, you may want to disable category filter in Magento's Layered Navigation block ("Shop By") in category view.

Sidebar menu block code

You can also display the sidebar menu inside the content of any CMS page or custom static block (and custom static block can be later added to any page through Magento widgets system – see this chapter for more details about adding custom content to selected pages: **4.8.1 Add blocks to selected pages**).

To display the menu, type in the menu code inside the static block content (Magento will automatically replace the code with the menu in the frontend), save the static block and flush Magento cache. Here's an example of the sidebar menu block code:

Important:

This block code has changed in version 1.4.0. If you used the code in previous versions, you need to replace the old code with the new one.

```
{block type="ultramegamenu/navigation" parent="X" depth="X" block_name="X"  
template="infortis.ultramegamenu/categories.phtml"}
```

There are several parameters that can help you configure the menu to suit your needs. Replace **X** with proper values of the parameters:

- **parent** – the menu will only contain children (subcategories) of the selected parent category. This parameter is similar to field **Parent Category** which is described earlier in this chapter. Available options:
 - **root** – show all top-level categories in the store.
 - **parent** – show the current category and its siblings (other categories from the same level as the current category).
 - **parent_no_siblings** – show the current category. Other categories from the same level as the current category will be ignored.
 - **current** – show subcategories of the current category.
 - Or specify the ID (an integer number) of any existing category.

- **depth** – specify how many levels in the hierarchy of categories (starting from the parent category) are to be included in the menu. This parameter is similar to [Maximum Depth](#) in the admin panel.
- **block_name** – the name of the block displayed as a heading.

Examples

Here you can see some examples of the block code:

```
 {{block type="ultramegamenu/navigation" parent="root" depth="2" block_name="All Categories in  
our store" template="infortis/ultramegamenu/categories.phtml"}}
```

```
 {{block type="ultramegamenu/navigation" parent="parent" depth="1" block_name="Current level of  
categories" template="infortis/ultramegamenu/categories.phtml"}}
```

```
 {{block type="ultramegamenu/navigation" parent="parent_no_siblings" depth="3"  
block_name="Current category with subcategories"  
template="infortis/ultramegamenu/categories.phtml"}}
```

```
 {{block type="ultramegamenu/navigation" parent="current" depth="4" block_name="Subcategories of  
current category" template="infortis/ultramegamenu/categories.phtml"}}
```

13.1.4 Category Labels

- **Label 1** – define the text of the category label 1. Labels can be assigned to every category in *Catalog > Manage Categories*.
- **Label 2** – define the text of the category label 2. Labels can be assigned to every category in *Catalog > Manage Categories*.

13.2 Drop-down boxes

13.2.1 Drop-down box configuration options

Each category in the menu can have a drop-down box containing subcategories or any other custom content. Below you can find a list of the drop-down box settings available in the admin panel. To configure the drop-down, go to *Catalog > Category Management*, select a category and open the *Menu* tab:

The screenshot shows the 'Fashion' category configuration in the 'Menu' tab. The left sidebar displays the category tree with 'Fashion' selected. The main panel contains the following configuration options:

- Submenu Type:** Mega drop-down
- Drop-down Width:** [STORE VIEW]
- Drop-down Content Proportions:** Left Block: 2, Subcategories: 8, Right Block: 2 [STORE VIEW]
- Number of Columns With Subcategories:** 4 [STORE VIEW]
- Category Blocks:**
 - Top Block:** Sample content of the block.
 - Left Block:** Sample content of the block.
 - Right Block:** Sample content of the block.
 - Bottom Block:** Sample content of the block.
- WYSIWYG Editor:** Buttons for each block.
- Category Label:** [STORE VIEW]
- Custom URL:** [STORE VIEW]

- **Submenu Type** – if category has subcategories, choose how subcategories should be displayed. You can choose the type individually for each category. For more details refer to **13.2.2 Submenu types**.

- **Drop-down Width** – in this field you can override the default width of the drop-down box. Width can be specified:
 - in pixels, for example: 150px,
 - or as a percentage, for example: 200% (percentage of the containing block's width).

In top-level categories (and only in top-level categories) if **Submenu Type** is “Mega drop-down” width of the drop-down box can also be specified in grid units (number between 1 and 12), for example: enter 4 and the drop-down box will have a width of 4/12 of the entire menu bar.

Default width of drop-downs:

- The mega drop-down of a top-level category is by default as wide as the menu bar.
 - The mega drop-down of a category on lower levels is by default as wide as the menu item of that category. So it is necessary to override the default width. For example, if you enter 300%, the drop-down will be three times as wide as the category item.
 - The classic drop-down is by default around 230 pixels wide.
- **Drop-down Content Proportions** – proportions between three sections of the mega drop-down box: **Left Block**, block of subcategories, **Right Block**.

Proportions are specified in grid units. For each section enter a number between 1 and 12. Sum of the grid units entered for all three sections has to be equal 12.

Note:

category blocks are displayed only in mega drop-down boxes so this option is applicable only for categories with “Mega drop-down” selected in the **Submenu Type** field.

- **Number of Columns With Subcategories** – in this field you can select the number of columns in which the subcategories are displayed in the mega drop-down box. For example, select 3 to display subcategories in three columns. Default value is 4.

Note:

subcategories are displayed in columns only in mega drop-down boxes so this option is applicable only for categories with “Mega drop-down” selected in the **Submenu Type** field (in the classic drop-downs subcategories are displayed one below the other).

- **Category Blocks** – if “Mega drop-down” was selected in the **Submenu Type** field, you can display additional content in the drop-down box.

Content can be inserted into four blocks which are described below. To distinguish these blocks from other blocks in Magento, we will call them **category blocks**. For more details refer to [13.2.3 Category blocks](#).

If you don't want to display any additional content in the drop-down box, simply leave these four fields empty:

- **Top Block** – category block displayed above the subcategories.
- **Left Block** – category block displayed at the left side of the subcategories.
- **Right Block** – category block displayed at the right side of the subcategories.
- **Bottom Block** – category block displayed below the subcategories.

Note:

if you need to split your content inside category blocks into columns, you can use the built-in grid system – refer to [6.4 Grid system](#) for more details about the grid.

- **Category Label** – eye-catching labels for categories. Labels needs to be defined in [System > Configuration > Menu > Category Labels](#). Refer to [13.1.4 Category Labels](#) for more details.
- **Custom URL** – enter a single hash character (#) to make the category **not clickable** in the main menu. It is especially useful when you want to create a drop-down box in the menu (to display some custom content) but don't want to redirect customers to any specific category.

Leave this field empty if no changes are needed and the category in the menu has to be a link to the category page.

You can also use this field **to replace the default link of the category** with any custom link. To do this, enter custom URL path into this field. The path will be appended to the base URL of the store to create a new link. For example, enter `contacts/` to create a link to Magento's default "Contact Us" page, or `customer/account/create/` to create a link to "Create an Account" page.

More info about adding custom links to the menu can be found in [13.3 Custom links in the menu](#).

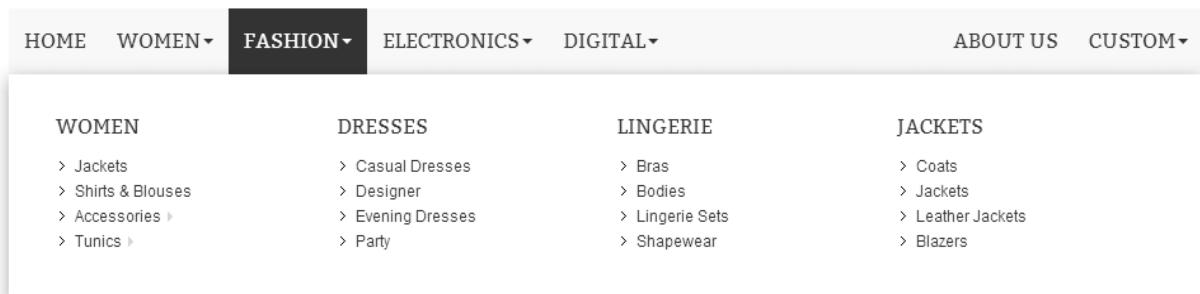
13.2.2 Submenu types

Available types

If the category has subcategories (i.e. it has a submenu), you can choose how the subcategories will be displayed. Go to *Catalog > Category Management*, select a category and open the *Menu* tab. There are three available options in **Submenu Type** field:

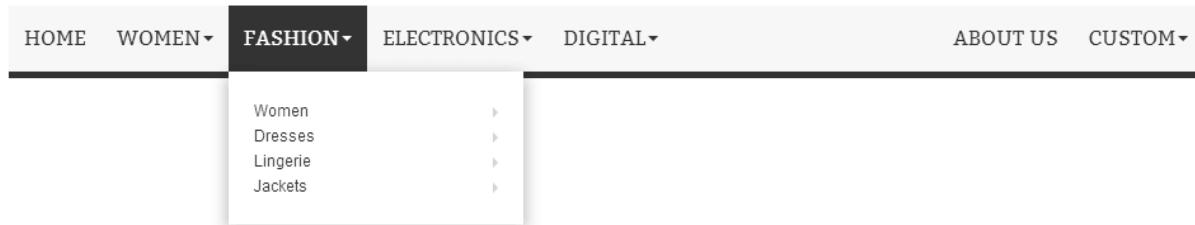
1. Mega drop-down

Wide drop-down box which can display subcategories in multiple columns. It can also display additional content from the category blocks (which are described later in this chapter).



2. Classic drop-down

Simple drop-down box which displays subcategories one below the other. It can display any number of levels of categories. Classic drop-down is the default type and it will be used if no type was specified for the category.



3. Simple submenu (no drop-down)

Additional type which can be used to display subcategories not in a drop-down box, but just below the category as a simple bulleted list.

Note:

this option can't be applied to top-level categories.

If in the previous example we will change the type of the “Dresses” category to “Simple submenu (no drop-down)”, the drop-down box will look like this:



So the subcategories of the “Dresses” category will be displayed as a simple bulleted list.

Multilevel menus

Multilevel menus are fully supported, categories inside a drop-down box can have their own subcategories which will also be displayed in drop-down boxes.

Additionally, all of the submenu types can be mixed together. For example, a classic drop-down can be nested inside a mega drop-down, and a mega drop-down can have classic drop-downs inside. See examples below:

The screenshot shows a navigation bar with links: HOME, WOMEN, FASHION (highlighted in blue), ELECTRONICS, DIGITAL, ABOUT US, and CUSTOM. The FASHION menu is open, displaying four main categories: WOMEN, DRESSES, LINGERIE, and JACKETS. The WOMEN category has a dropdown menu with items: Jackets, Shirts & Blouses, Accessories, and Tunics. The Tunics item has a nested dropdown with sub-items: Girls, Women, Long, Medium, and Short. The DRESSES category has a dropdown menu with items: Casual Dresses, Designer, and Evening Dresses. The LINGERIE category has a dropdown menu with items: Bras, Bodies, Lingerie Sets, and Shapewear. The JACKETS category has a dropdown menu with items: Coats, Jackets, Leather Jackets, and Blazers.

The screenshot shows a navigation bar with links: HOME, WOMEN, FASHION (highlighted in black), ELECTRONICS, DIGITAL, ABOUT US, and CUSTOM. The FASHION menu is open, displaying four main categories: Women, Dresses, Lingerie, and Jackets. The Women category has a dropdown menu with items: Dresses, Lingerie, and Jackets. The Dresses item has a nested dropdown with sub-items: Jackets, Shirts & Blouses, Accessorie, and Tunics. The Accessorie item has a nested dropdown with sub-items: Bags, Shoes, and Jewelry. The Bags category has a dropdown menu with items: Bags, Handbags, Purses, and Shoulder Bags. The Shoes category has a dropdown menu with items: Flat Shoes, Flat Sandals, Boots, and Heels. The Jewelry category has a dropdown menu with items: Bracelets, Necklaces & Pendants, Pins & Brooches, and Watches.

13.2.3 Category blocks

If “Mega drop-down” was selected as the submenu type, you can display additional content inside the drop-down box. Custom content can be inserted into four **category blocks**. On the example below, blocks of the “Fashion” category are marked with light red:

Top Block

Left Block

| | | | |
|--|--|--|--|
| WOMEN | DRESSES | LINGERIE | JACKETS |
| > Jackets > Shirts & Blouses > Accessories » > Tunics » | > Casual Dresses > Designer > Party > Vests | > Bras > Bodies > Lingerie Sets > Shapewear | > Coats > Jackets > Leather Jackets > Blazers |

Right Block

Bottom Block

Mega drop-downs can also display category blocks of lower categories. For example, if the subcategories of the “Fashion” category (“Women”, “Dresses”, “Lingerie” and “Jackets”) have any content inside Top Block and Bottom Block, these blocks will be displayed in the drop-down. On the example below, these blocks are marked with light blue color:

Top Block

Left Block

| | | | |
|--|--|--|--|
| WOMEN | DRESSES | LINGERIE | JACKETS |
| > Jackets > Shirts & Blouses > Accessories » > Tunics » | > Casual Dresses > Designer > Party > Vests | > Bras > Bodies > Lingerie Sets > Shapewear | > Coats > Jackets > Leather Jackets > Blazers |

Top Block

Bottom Block

Top Block

Bottom Block

Top Block

Bottom Block

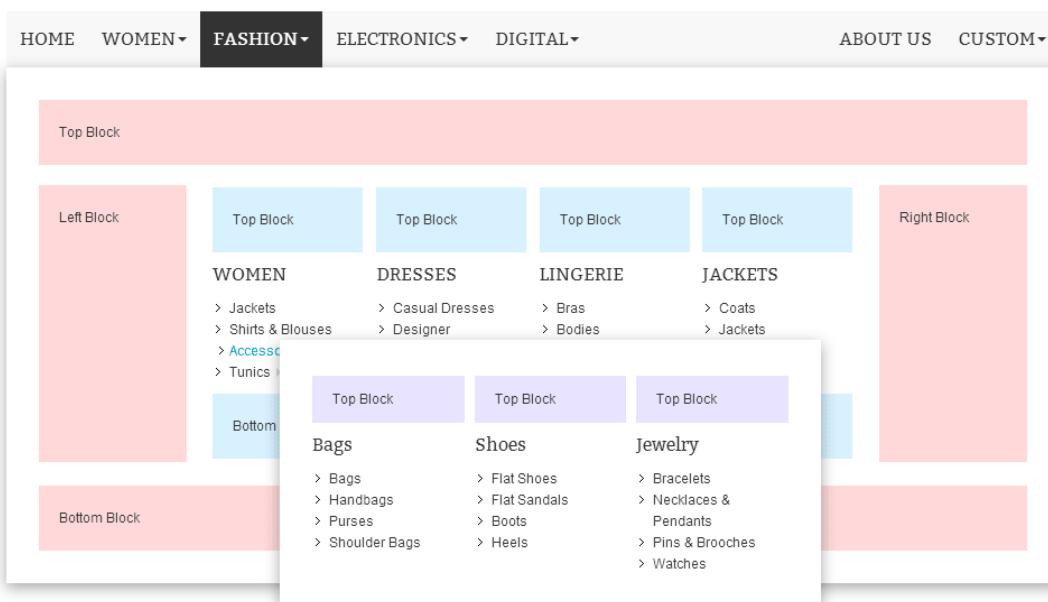
Top Block

Bottom Block

Right Block

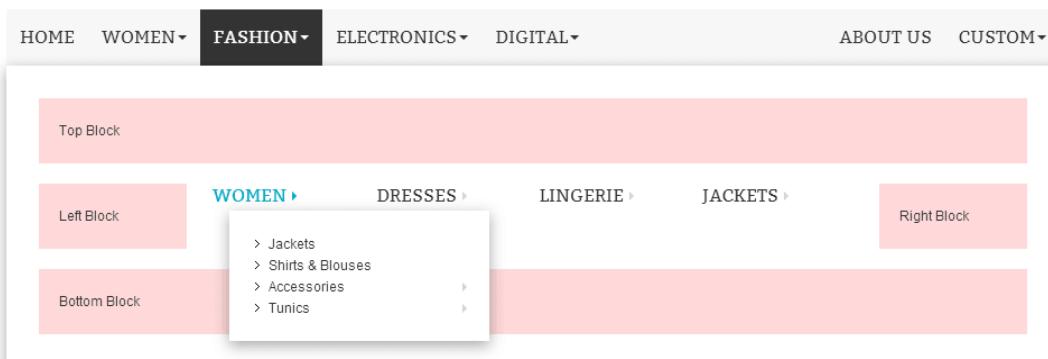
Bottom Block

Moreover, subcategories of the “Fashion” category can have their own drop-down boxes. For example, if the “Accessories” category is configured as a mega drop-down (i.e. “Mega drop-down” was selected in the **Submenu Type** field), then a drop-down box will be displayed on mouse hover over the “Accessories” category. That drop-down can also contain its own category blocks with custom content (on the example below, these blocks are marked with light purple color):



Note that in this example, categories “Women”, “Dresses”, “Lingerie” and “Jackets” can't have “Mega drop-down” selected in the **Submenu Type** field (if it was selected, it will be ignored) because their subcategories are already displayed in the mega drop-down.

But categories “Women”, “Dresses”, “Lingerie” and “Jackets” **can have** “Classic drop-down” selected in the **Submenu Type** field. If they are configured this way, menu will look like this:

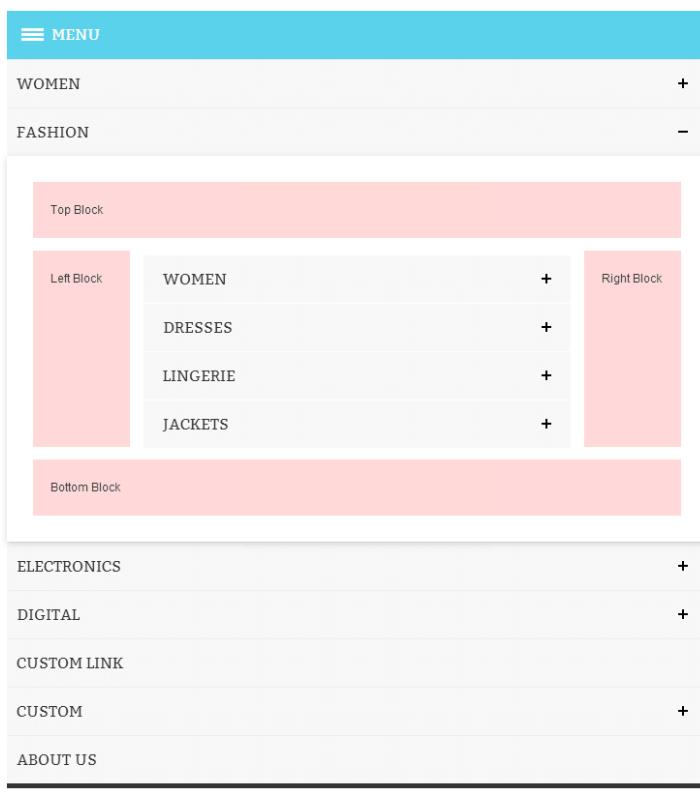


13.2.3.A Category blocks in mobile menu

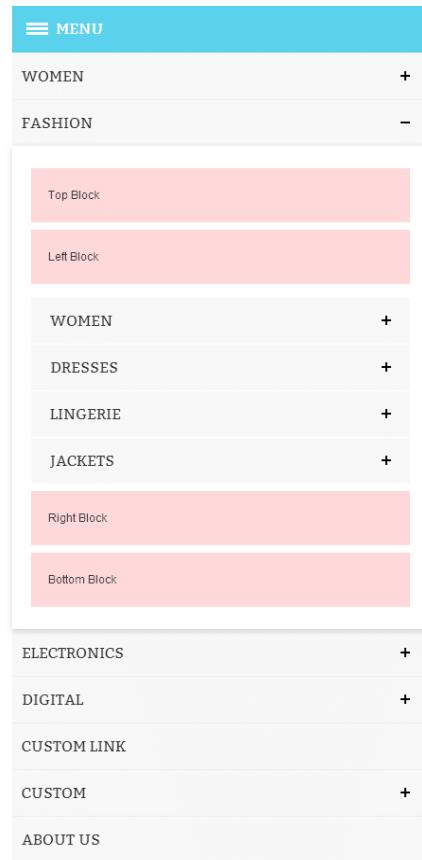
If the mobile menu is enabled, category blocks will also be displayed in the mobile menu (picture 1.). Mobile mode is activated when browser viewport is narrower than the threshold specified in the menu settings.

If screen is narrower than 768 pixels, category blocks will be displayed one below another (picture 2.), even the **Left Block** and the **Right Block**.

Picture 1.



Picture 2.



To hide category blocks in the mobile menu, you can use the **Show Category Blocks From X Levels of Categories** option which can be found in *System > Configuration > Menu > Mobile menu*. Refer to **13.1.2 Mobile Menu** for details.

You can also hide selected portions of the content inside the category blocks. There are two special classes:

- `hide-in-mobile-menu` – this class will hide the element when the menu is in the mobile mode
- `hide-in-desktop-menu` – this class will hide the element when the menu is in the standard mode.

Example below shows how to use these classes. Add this code inside one of the category blocks, e.g. **Top Block**:

```
<div class="hide-in-mobile-menu">  
    This text will not be visible in mobile menu  
</div>  
  
<div class="hide-in-desktop-menu">  
    This text will not be visible in desktop menu  
</div>
```

13.2.4 Sample content for category blocks

Here you can find some samples of the content for category blocks. They are based on the samples from the demo.

Category "Women"

Content of the *Right Block*:

```
<a href="{{store direct_url='dress7.html'}}">
    <div class="ban ban-effect-3 ban-caption-fade-out">
        
        <div class="cap" style="background-color: rgba(0,0,0, 0.2);">
            <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                <span class="ic ic-gift ic-6x animated infinite pulse"></span>
            </div>
        </div>
    </div>
</a>
```

Category "Fashion"

Content of the *Right Block*:

```
{block type="ultimo/product_list_featured"
template="catalog/product/list_featured_slider.phtml"
category_id="3" product_count="4" breakpoints="[0, 1], [480, 2], [768, 1], [960, 2]"
timeout="4000" move="1" centered="1" size="size-s" hide_button="1" img_width="110"
block_name="Recommended Bags"}
```

Category "Electronics"

Content of the *Top Block*:

```
<div class="links-wrapper-separators">
    <ul class="links">
        <li class="label hidden-xs hidden-sm">Recommended:</li>
        <li>
            <a href="{{store
```

```
direct_url='electronics/smartphones/phone3.html'}})">Configurable</a>
    </li>
    <li>
        <a href="#">Galaxy S3</a>
    </li>
    <li>
        <a href="#">Galaxy S4</a>
    </li>
    <li>
        <a href="#">iPhone 5</a>
    </li>
</ul>
</div>
<div class="links-wrapper-separators pull-right hide-in-mobile-menu hidden-xs hidden-sm">
    <ul class="links">
        <li>
            <a href="#">Add</a>
        </li>
        <li>
            <a href="#">Custom</a>
        </li>
        <li>
            <a href="#">Links</a>
        </li>
    </ul>
</div>
<span class="section-line"></span>
```

Content of the *Left Block*:

```
<a class="hidden-xs" href="{{store direct_url='electronics/phone-accessories.html'}}">
    <div class="ban ban-caption-hinge-left">
        
        <div class="cap" style="background-color: #ffc60f;">
            <div class="cap-center-vertically cap-center-horizontally cap-no-bg">
                <span class="ic ic-star ic-4x animated infinite pulse margin-bottom"></span>
                <h3 class="no-margin">Go to category</h3>
            </div>
        </div>
    </div>
</a>
<br/>
<h3>Featured Category</h3>
<p>This is a custom block ready to display any content. You can add blocks to any category in the catalog...</p>
<a class="button" href="#">View Category &ampnbsp<span class="ic ic-arrow-right"></span></a>
```

Category "Phones & Tablets"

Content of the *Top Block*:

```
<a href="{{store direct_url='electronics/smartphones.html'}}">
    <div class="ban ban-caption-hinge-up">
        <span class="image ic ic-3x ic-smartphones ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
        <div class="cap">
            <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                <h4 class="no-margin">Phones & Tablets</h4>
            </div>
        </div>
    </div>
</a>
```

Category "Accessories"

Content of the *Top Block*:

```
<a href="{{store direct_url='electronics/phone-accessories.html'}}">
    <div class="ban ban-caption-hinge-up">
        <span class="image ic ic-3x ic-callcenter ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
        <div class="cap">
            <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                <h4 class="no-margin">Accessories</h4>
            </div>
        </div>
    </div>
</a>
```

Category "Cameras"

Content of the *Top Block*:

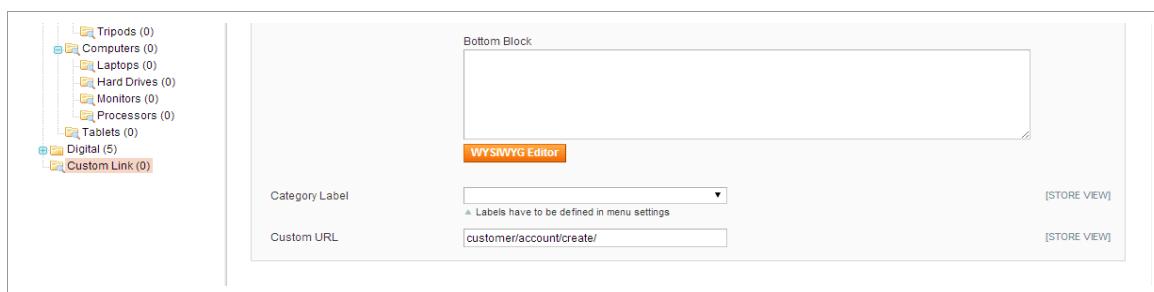
```
<a href="{{store direct_url='electronics/cameras.html'}}">
    <div class="ban ban-caption-hinge-up">
        <span class="image ic ic-3x ic-gift ib ib-hover ib-square ib-size-xxl"
style="background-color: #333;"></span>
        <div class="cap">
            <div class="cap cap-center-vertically cap-center-horizontally cap-no-bg">
                <h4 class="no-margin">Cameras</h4>
            </div>
        </div>
    </div>
</a>
```

13.3 Custom links in the menu

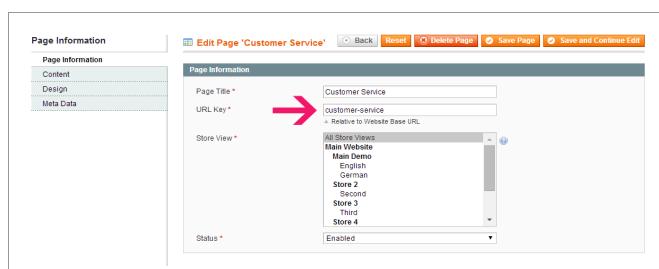
The main menu typically includes the selection of categories from the catalog. But you can also add links to content pages and external websites. In order to add a link, follow the steps described in the example below.

Let's say you want to create a link to Magento's "Create an Account" page.

1. Go to *Catalog > Category Management* and create a new category in your catalog by clicking the **Add Subcategory** button.
2. Set **Is Active** to "Yes".
3. The name of the category will be the name of the link so you can use any name you like.
4. Open the *Menu* tab. In the **Custom URL** field, enter URL path of the "Create an Account" page which is `customer/account/create/`. This path will be appended to the base URL of the store to create a link to that page:



To create a link to some other page you just need to enter proper URL path. For example, to create a link to a CMS page, in the **Custom URL** field, enter the **URL Key** of that page. To check what is the **URL Key** of a page, go to CMS > Pages, select a page and find the field:



- Finally, go to your store and make sure the new link appears in the main menu. If it's not, you may need to reindex Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**).

Alternatively, to add custom link to the menu, you can use the method (which uses URL rewrites) described in this article: <http://www.magentocommerce.com/knowledge-base/entry/adding-page-links-in-the-navigation-bar>

13.3.1 Custom links inside a static block

Another method to add custom links to the main menu, is to create a static block (in *CMS > Static Blocks*) with identifier `block_nav_links`. Content of this block will be displayed in the menu bar so you can quickly add multiple links to the menu.

To display links properly, content of the block should have the following structure:

```
<li class="nav-item level0 level-top right">
    <a class="level-top" href="{{store direct_url='about-magento-demo-store'}}">
        <span>Sample Internal Link</span>
    </a>
</li>
<li class="nav-item level0 level-top right">
    <a class="level-top" href="http://infortis-themes.com/">
        <span>Sample External Link</span>
    </a>
</li>
```

Important:

note that there are no `` tags at the beginning and at the end of this code snippet. This code will be inserted inside the menu bar HTML which already contains `` tags and additional tags would break the structure of the menu.

You can remove class `right` if you don't want the links to float to the right side of the menu bar.

In version 1.9.0 a new class `nav-item` was added to the menu items ``. You will need to update the content of your block and add that class to display the links correctly.

Also, you may need to disable Magento's WYSIWYG editor, otherwise it will add `` tags automatically. It is recommended to disable the WYSIWYG completely so that it doesn't process the code when you don't need it. Turn WYSIWYG on only when it is needed. Go to *System > Configuration > Content Management* and set it as *Disabled by Default*.

For more information about CMS **markup tags**, such as
`{{store direct_url='about-magento-demo-store'}}`, refer to **4.8.3 CMS markup tags**.

13.4 Custom drop-down boxes in the menu

Drop-down boxes with custom content can be easily added to any top-level category in the main menu. In order to add a drop-down, follow the steps described in the example below.

1. Go to *Catalog > Category Management* and create a new category in your catalog by clicking the **Add Subcategory** button.
2. Set **Is Active** to **Yes**.
3. The name of the category will be the name of the category item in the menu so you can use any name you like. Drop-down box will be displayed on mouse hover over that item.
4. Open the *Menu* tab. In the **Submenu Type** field, select “Mega drop-down”.
5. In the **Top Block** field, enter the content which you want to display in the drop-down box. It can be any HTML or just a simple text – whatever you need:

The screenshot shows the 'Custom Block (ID: 142)' configuration page. On the left, there's a sidebar with 'Categories' navigation, including 'Add Root Category' and 'Add Subcategory' buttons. Below that is a 'Choose Store View' dropdown set to 'All Store Views'. Under 'Default Category (0)', there are links for 'Women (12)', 'Fashion (28)', 'Electronics (17)', and 'Custom Block (0)'. The main content area has tabs: 'General Information', 'Display Settings', 'Custom Design', 'Menu' (which is selected), and 'Category Products'. In the 'Menu' section, the 'Submenu Type' is set to 'Mega drop-down'. The 'Top Block' field contains the following HTML:

```
<h3>Sample heading</h3>
<p>This is a custom block ready to display any content</p>
```

Below the 'Top Block' is a 'WYSIWYG Editor' button. There are also sections for 'Drop-down Width', 'Drop-down Content Proportions', 'Number of Columns With Subcategories', and 'Category Blocks' (with a 'Left Block' field).

6. In the **Custom URL** field, enter a single hash character (#) to make this category not clickable (or leave this field empty if no changes are needed and the category in the menu has to be a link to the category page).

7. Finally, go to your store and make sure the new item appears in the menu. If it's not, you may need to reindex Magento (**4.2 Index management**) and flush the cache (**4.1 Caching system**).

Static block as a drop-down box

Another method to add custom drop-down box to the main menu, is to create a static block (in *CMS > Static Blocks*) with identifier `block_header_nav_dropdown`. Content of this block will be displayed in the menu as a drop-down box. Title of that block will be used as the name of the menu item.

To align this menu item to the right side of the menu bar, use option **Custom Static Block - Align Right** which can be found in *System > Configuration > Menu > Top Menu*.

14. Troubleshooting

14.1 General issues after installation

14.1.1 Some features don't work or give errors

Here's the list of possible sources of problems:

- a) Magento cache (or external cache) was not flushed after installation of the theme.
- b) CSS files which are responsible for the visual appearance of the theme were not generated automatically. Go to *System > Configuration > Ultimo > Theme Design* and click **Save Config** button. Then go to *System > Configuration > Ultimo > Theme Layout* and click **Save Config** button again. Then, flush Magento cache and refresh your browser's cache.
- c) You can see all sorts of error messages if the **Compilation** mode was not disabled before installation.
- d) Some of the theme files were not uploaded or are broken. Re-upload original theme files to your server (overwrite existing files) and flush the cache. Note that files can get broken if you send them in large numbers via FTP without zipping. So it is highly recommended to upload zipped files.
- e) File permissions issue – refer to [14.1.4 File permissions](#) for more details.
- f) Theme wasn't enabled properly. Theme should only be enabled in *System > Configuration > Design* (the same as it was described in the installation steps). Refer to [4.3 How to enable the theme](#) for more details.
- g) You have some third-party extensions which change/break theme's default behavior or cause some conflicts. Disable all extensions and flush the cache, then enable extensions one by one to check which one is causing problems. Refer to [2.3 Third-party extensions](#) for more details about third-party extensions.
- h) You edited theme files and accidentally removed or changed something in the files. Re-upload original theme files to your server (overwrite existing files) and flush the cache.
- i) You have customized the theme by creating custom sub-theme and your

modifications override or break the default behavior of the theme.

- j) You have some elements that left from your previous theme (e.g. in Magento's "base" theme in `app/design/frontend/base/`) and that breaks the new theme.

Also check your log files in `var/log/` (log settings should be enabled in `System > Config > Developer`) for more information about possible errors.

14.1.2 Blank page or 404 not found – when trying to open theme admin pages

Log out from Magento admin panel, delete all files in `var/cache/` directory (to clear Magento cache), and log in again.

14.1.3 There is no theme admin tab in the admin panel, or I get a blank page, or Access Denied error

Log out from Magento admin panel, delete all files in `var/cache/` directory, clear web browser cache/cookies for your store domain and log in again.

14.1.4 File permissions

Incorrect file permissions on your server may cause many issues. After uploading any files to your server in order to install themes or extensions you need to set correct file permissions for all the uploaded files – the same permissions as for the default Magento files.

File permissions may differ depending on server – usually 755 for folders and 644 for files. In case of any doubts about permissions, please contact your hosting provider. Refer to this article for more details:

http://www.magentocommerce.com/wiki/1_-_installation_and_configuration/magento_filesystem_permissions

In case of problems related to images refer to: **14.1.10 Images are not displayed**

14.1.5 Theme features don't work after installation or upgrade

If some theme features don't work after installation or some resources (e.g. images) are not displayed in the frontend, it often indicates problems with file permissions – your

Magento doesn't have access to the theme files because of incorrect file permissions on your server. For more details refer to [14.1.4 File permissions](#)

After theme upgrade

If you upgraded the theme in your store and you're using custom sub-theme, you will also need to **upgrade your sub-theme**. This is extremely important step: review all the files in your sub-theme to make sure they don't override new features added to the theme.

For example: if the product page template file was changed in the update (e.g. a new option was added: ability to display product SKU), you will need to apply that change in your sub-theme (in the product page template file). Otherwise SKU will not be displayed in your store because Magento will use the template file in your sub-theme to render the product page and your template file doesn't have the new feature. So you will need to add the new feature manually (copy the new code from the original template file to the file in your sub-theme).

14.1.6 Fatal error: Class 'Mage_Default_Helper_Data' not found

If you see the following error message:

```
Fatal error: Class 'Mage_Default_Helper_Data' not found in ***/app/Mage.php on line 546
```

(where ******* stands for the directory of Magento on your server) it indicates incorrect installation:

- Magento's **Compilation** mode was not disabled in *System > Tools > Compilation* before theme installation,
- or the theme wasn't enabled properly. Theme should only be enabled in *System > Configuration > Design* (the same as it was described in the installation steps). Refer to [4.3 How to enable the theme](#) for more details.

14.1.7 "Package with this name does not exist" message is displayed when trying to enable the theme

This message means that:

- a) theme files were not uploaded,
- b) or they are not in the correct place,
- c) or there is a problem with file permissions on your server – refer to chapter [14.1.4](#)

[File permissions](#) for more details.

14.1.8 After theme installation the store gives blank page or an error

Log out from Magento admin panel. Change the owner of the uploaded files to the web server user and set correct file permissions for them. Delete all files in `var/cache/` directory (do not delete the directory) and log in again.

14.1.9 Tabs, sliders or other interactive elements don't work

This may be caused by jQuery conflict: one of your custom extensions is loading another version of jQuery. You need to remove that version of jQuery or remove that extension.

14.1.10 Images are not displayed

Sometimes after installation some images on custom CMS pages or inside static blocks (e.g. images in the slideshow) are not displayed. First, check if images were uploaded and if correct paths to those files can be found in the source code of your page (click `CTRL+U` in most browsers).

Note:

in fresh Magento installation, the default sample home page tries to load sample images which doesn't exist in this theme (so you will see blank rectangles instead of images). Go to `CMS > Pages > Home page`, remove the default content or replace it with your own content.

If all files were uploaded and paths are correct, it means that your Magento doesn't have access to those images because of incorrect file permissions on your server. You need to set correct file permissions for all images and folders containing images (in most cases images are stored in Magento's `media` directory). Refer to chapter [14.1.4 File permissions](#) for more details.

14.1.11 Sliders: previous/next buttons don't work in some browsers

Please see this thread on Flexslider forum (Flexslider is a jQuery plugin used for sliders), some users reported that the issue can be caused by a Wacom Driver, they were able to

resolve the issue by unplugging their tablets:

<https://github.com/woothemes/FlexSlider/issues/589#issuecomment-14804424>

You may also want to reinstall your web browser and remove all plugins to check if some of them can cause conflicts.

14.1.12 Slideshow is not displayed on the home page

Make sure the configuration of the slideshow is correct. Enable all the static blocks which are used as slides in the slideshow. Finally, flush the cache.

If this doesn't help:

- Non-standard value in *System > Configuration > Web > Default Pages > Default Web URL*. Set the default value: `cms`
- Some custom extensions change the default behavior of Magento's home page, so the slideshow module is not able to load on the home page anymore. How to recognize that case: open the source code of the home page in your web browser (`CTRL+U` in most browsers). The `<body>` tag on the home page should have CSS class `cms-index-index` (this is Magento's default class). If you don't see this class, then it is probably that case, and you will need to disable the extension which is changing the default behavior of Magento's home page.

14.2 Configuration

14.2.1 Theme configuration changes do not appear on the frontend

Flush Magento cache in *System > Cache Management*, clear web browser cache/cookies for your store domain and refresh the page in your web browser.

Note:

If you created a sub-theme to customize the theme, do not copy folder `skin/frontend/<ThemeName>/default/css/_config/` (and files stored inside) to your sub-theme. If you do this, files inside this folder will override your theme settings.

14.2.2 "Failed generating CSS file" message after saving theme configuration

What can cause the problem:

- You didn't upload all theme files to the following directory:
`app/design/frontend/base/default/template/infortis/ultimo/css`
(you should see there three files: `design.phtml`, `grid.phtml`, `layout.phtml`)

Or your Magento doesn't have access to those files because of incorrect file permissions on your server. Refer to chapter **14.1.4 File permissions** for more details.

- Your Magento doesn't have permissions to write in the following directory:
`skin/frontend/ultimo/default/css/_config`
Set correct file permissions for that directory. Refer to chapter **14.1.4 File permissions** for more details.

14.3 Third-party extensions

14.3.1 One of my extensions doesn't work with this theme

This theme may not be compatible with some third-party extensions. It's impossible for theme author to make the theme compatible out of the box with all existing extensions because there are thousands of available Magento extensions.

Generally all extensions can work with all themes but sometimes they have to be integrated manually. That's because sometimes the extension and the theme override the same Magento template file so there can be a conflict. In such cases you need to customize the theme and merge two conflicting template files into one template file.

This isn't something specific to our themes, this is general rule for all Magento themes and extensions.

Refer to chapter [2.3 Third-party extensions](#) for more details.

14.3.2 One of the features stopped working after third-party extension was installed

There can be a conflict, if the extension and the theme override the same Magento template file. You should contact author of the extension for more information. Also, check your log files in `var/log/` (log settings should be enabled in *System > Config > Developer*) for more information about the error. Refer to chapter [2.3 Third-party extensions](#) for more details about third-party extensions.